International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

Malicious Website Detection using Machine Learning

Liudmila Swati Xess[1]
B.Tech-CSE
Sharda University
Greater Noida-UP

Mahek Khera[3]
B.Tech-CSE
Sharda University
Greater Noida-UP

Tanishka Prasad[2]
B.Tech-CSE
Sharda University
Greater Noida-UP

Rupender Singh[4]
B.Tech-CSE
Sharda University
Greater Noida-UP

Ms. Manpreet Kaur Aiden[5]
Assistant Professor, SET
Sharda University
Greater Noida-UP

***Abstract* — Malicious means an intent of doing harm. A malicious website intends to cause harm to the end user by spreading malware, infecting the victim's system and stealing critical information. The worldwide lockdown in the year 2020 saw an increase and shift to internet services being used as a mode to run operations while staying at home. This, in turn resulted in an increasing number of cybercrimes by cyber criminals and huge data losses by companies.. To stop these assaults, malicious URLs must be located and threat kinds must be identified. Because malicious web pages import exploits from distant resources and hide exploit code, static properties describing these behaviours can be utilised to identify the vast majority of malicious web pages. In past years, several methods and models have been proposed to identify such Phishing URLs. In this paper we review the previous studies and propose a machine learning approach to detect malicious websites using the machine learning model with best accuracy. Moreover, we also perform a reconnaissance on the URL to provide additional information like port status, directories and subdomains associated with the website.**

Keywords: Malicious website; phishing; cybercrimes; machine learning.

I. INTRODUCTION

As the technology has advanced and grown , more and more services have become available on the internet and web applications have made them accessible to a larger number of people. It is used for various tasks like banking, shopping, diversions, asset transfer, news, and long-distance interpersonal interactions. However, as these activities that help people in their daily lives become increasingly entwined with the Internet, the web's development has rewarded the digital hoodlums. With this development, the malware situation has also changed tremendously, becoming stealthier and polymorphic rather than harming machines. The majority of malware is designed to either steal the victim's personal information or force the victim's computer to join a malware distribution network. The web is a common method for spreading malware; attackers take advantage of flaws in web browsers, web applications & operating systems to gain access to a victim's computer, which is then utilised for malicious operations like load splash, botnets, keyloggers, spamming, DDOS attacks and so on. These malicious websites do not only steal or harm clients' data, but also allow programmers to control the infected computers. It reaches a point where numerous online wrongdoings are condoned. Phishing assaults occurring today are complex and progressively more challenging to recognize. A review led by Intel viewed that as 97% of safety specialists come up short at recognizing phishing messages from certified messages [27].

Coding languages such as HTML and JavaScript are commonly used to represent website pages in online web applications. Other methods for downloading and executing code from the Internet include Adobe Stream, and visual important content. Similarly, most web applications have a module component that allows outsiders to extend the program's functionality. While these codes are useful to web application developers, attackers can use source codes produced in these dialects to generate new forms of noxious site pages. Clients who browse the vulnerable website pages, in other words, may become assault survivors. The aggressor can obtain the client's basic info from the PC and use the contaminated PCs to barter for more PCs belonging to the same group. As a result, pinpointing the specific position of malicious internet pages and preventing the emergence of new types of detrimental pages is critical.

We can keep our personal and professional data confidential, secure, and accessible by identifying malicious URLs. A popular countermeasure is avoiding bad URLs, which can be generated from a variety of sources. Boycotting has no false positives, but it is only effective against bad URLs that have been identified. It can't tell the difference between cryptic spiteful URLs and those that aren't. To protect our data, we need a more efficient and effective means of determining a phishing url.

Getting information about anything is called reconnaissance. It plays an important role in deciding whether a domain/link

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

is malicious or not.In this process, a couple of key information points are collected about the domain/link so that we can assess on the basis of them whether the provided domain/link is a legitimate one or a fake one. The basic information that is collected during the reconnaissance process is the state of ports on the server on which the domain is hosted, the number of subdomains that particular domain/link has and the kind of directories that domain has. All these three things have very significance as all of these three features have a very different value in case of a fake or a phishing domain.

II. RELATED WORK

Author [1] suggested a solution where execution of the proposed identification technique is compared with the other detection strategies. SVMs have been used as supervised learning classifiers in the misuse detection model that needs labelled training data sets. It improved detection accuracy to 98.9%.

Author [2] suggested an algorithm based on the URL lexical and the page content features Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), K Nearest-Neighbour (KNN) using benign set- web search and Alexa website ranking. It achieved 97% accuracy showing that combination of the feature groups has shown the higher true positive rate.

Author [3] used an architecture of sparse random projection, logistic regression & DL model. Stacked denoising auto-encoders were utilized to separate significant level highlights; logistic regression as a classifier was utilized to bunch them as malevolent/harmless. Over 27,000 labelled samples & accuracy of 95%, with a false positive rate under 4.2% in the best case.

Author [4] suggested creating a list of blacklisted domains, IP addresses and Urls and whenever a person receives a mail from that domain, URL or IP it is shown as Malicious. The dataset used in this was Human Feedbacks / Blacklisted IP's. The result that came out was that malicious website detection can be done in real time from a given list of IP addresses, URLs, domains.

Author [5] combined different feature sets and feature values, dynamically taking snapshot of webpage execution , timely update the set of feature type and feature values , building richer set of features , proper characterization of attack payloads , drawing line between stable features and dynamically changing feature .Major finding in this research was The feature set and feature values.

In the solution given by Author [6], SVM is utilized to identify pernicious URLs. Two multi-mark arrangement strategies, (RAkEL and ML-kNN]), are utilized to distinguish assault types. The dataset used in this are Benign URLs, Spam URLs, Phishing URLs, Malware URLs. The finding was that this method has an accuracy of 98.2%.

Author [7] proposed a solution with feature extraction and used an online learning method. The result came out that this solution is 97% accurate.

Author [8] proposed an algorithm which plans two sorts of elements for web phishing: unique highlights and communication highlights. An identification model in light of Deep Belief Networks (DBN) is then introduced. In this they had the option to accomplish a roughly 90% genuine positive rate and 0.6% false positive rate.

Author [9] used models like Decision Tree, Ada-Boost, Logistic Regression, KNN, Random Forest, Gradient Boosting, Support Vector Machine, Neural Networks, and XGBoost. The PhishTank dataset consisting of 6157 real sites and 4898 phishing sites was utilized. The outcome that came out was In KNN grouping we figured out the best execution is gained when we set k to 5.

In the algorithm proper by Author [10] ANOVA (Analysis of Variance) test and XGBoost (eXtreme Gradient Boosting) calculation are utilized to recognize the 17 most significant elements. At last, the dataset is utilized to become familiar with the XGBoost classifier. 41 highlights of malignant URLs were removed from the information cycles of space, Alexa and obfuscation. By this algorithm they were able to achieve 99.98%.

In the solution proposed by Author [11], the three algorithms used for classification are Logistic Regression, Naive Bayes, and Decision Forest. Each algorithm was evaluated with a large dataset, and then tested with a single URL from the smartphone. All classifiers reported 99.8% accuracy.

The author [12] analyzed the URL for various features. On the basis of these factors a score was provided and if it comes out to be less than a certain no than that URL will be considered to be a phishing URL. By applying this algorithm, the effectiveness had increased up to 99.1 %.

In the paper by author [13], a versatile grouping of pernicious web code by machine learning approach like Naïve-Bayes, SVM and KNN algorithm for detecting the exploitation of user inputs has been proposed. The models have shown accuracy of 98.60, 98.88 and 98.60 respectively.

Author [14] has used related highlights of pictures, edges and text of genuine and non-authentic sites and related man-made reasoning calculations to identify web phishing. This approach showed an accuracy of about 98.3%.

In the next paper by author [15], he has analysed the characteristics of a malicious web page systematically and presented important features for machine learning. The algorithms used include Decision Tree, Naïve Bayes, Boosted Decision Tree & SVM with respective accuracies of 58.28, 94.74, 93.52 and 96.14.

The paper by author [16] compares the outcomes of a variety of machine learning classification approaches, including Random Forest (RF), Logistic Regression (LR), K-Nearest

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

Neighbours (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Stochastic Gradient Descent (SGD), , and Decision Tree (DT). To detect dangerous websites from the OpenPhish domain, the best performing classifier is employed.

Author [17] used the Scikit Learn Package to implement a Multilayer Perceptron, Random Forest Classifier, Logistic Regression, and Decision Tree Classifier package for machine learning algorithms. In this, each has a tokenized dataset and a typical train and test dataset, and looking at every calculation gives a slight contrast in outcome precision.

In paper by author [18] the detection model is made up of numerous components, including Malicious URL Acquirement Module, Topic Analyzer, Web-page Analyzer, Comprehensive Analysing & Labelling Module, Attacks Classifying Module & Output.

In the solution proposed by Author[19] , The URL features are discriminated on the basis of 4 parameters i.e Feature Analysis , Feature Semantics, Feature Fingerprinting feature , Feature Fusion. The Accuracy came out to be 99.89% .

Author [20] proposed a solution in which the URL selection and extraction is done on 3 basic categories: Host based Features, Lexical features, and Content Based features. The algorithms used to process the data are SVM and Rf

In the solution given by the author [21] . A novel capsule based neural network consisting of 4 branches where 1 convolution layer and 2 capsule layers are used to decide whether the URL is legitimate or a phishing URL. The output of all the 4 layers is averaged out to improve the generalization of the approach.

Author[22] proposed a methodology in which logistic regression is used which includes a dependent variable which can be represented in binary (0 or 1) . The dataset used to feed the algorithm contains different features of an URL on the basis of which the Algorithm decides which URl is legitimate and which is not. The result came out to be 98.42%.

Author [23] recommends a new malevolent URL detection technique in view of a deep learning model to safeguard against web assaults with a success rate of 99.14%.

In the next paper by author [24], machine Learning methods are utilized for detection of phishing sites in view of lexical elements, host properties and page significance properties. Models included Naive Bayes, J48, IBK and SVM with accuracy of 68.60, 93.20, 88.30 and 83.93 respectively.

In our last paper by author [25],a multi-faceted element phishing recognition approach in view of a quick identification strategy by utilizing deep learning (MFPD) is proposed, which can decrease the detection time for setting a threshold. The success rate of the approach is 98.9

TABLE I. LITERATURE SURVEY

Ref No.	Dataset	Methodology	Accuracy
[1]	-Custom	-Single misuse detection method using the decision tree algorithm -Single anomaly detection method using a one-class SVM	98.9%
[2]	-Benign set- web search -Alexa website ranking verified by Google safe browsing -Common public announced malware and exploited websites	-Artificial Neural Network (ANN) -Naive Bayes (NB) -K Nearest-Neighbor (KNN) -Decision Tree (DT) -Support Vector Machine (SVM)	97%
[3]	-VX Heaven -Alexa Top Sites -Malicious Web Site Labs	-Deep learning model -Logistic regression. -Sparse random projection	95%
[4]	-Human Feedbacks -Blacklisted IP's	-Create a list of blacklisted domains, IP addresses and URLs and whenever a person receives a mail from that domain, URL or Ip it is shown as Malicious	--
[5]	-Online Records -Feature Identification -Feature values	- Combine different feature sets and feature values, update, build & characterize attack payloads, drawing line between stable features and dynamically changing features	--
[6]	-Benign URLs -Spam URLs -Phishing URLs -Malware URLs	-Support Vector Machine (SVM) -Two different multi label classification methods i.e, RAkEL and ML-KNN	98.2%
[7]	Malicious URLs and normal URLs, which are used for training and testing classifiers	-Online learning	97%

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

[8]	- Ip Flows Collected from the Internet Service Provider	-Deep Belief Networks (DBN)	90%
[9]	-6157 Genuine/Legitimate Websites combined with 4898 phishing websites (Name of the dataset: Phishtank)	-Logistic Regression Model ,Decision Tree, Random Forest, Ada-Boost, Support Vector Machine(SVM) , KNN, Neural Networks, Gradient Boosting, XGBoost	--
[10]	-Alexa	-(eXtreme Gradient Boosting) XGBoost algorithm -(Analysis of Variance) ANOVA test	99.98%
[11]	-URLS from large dataset and ability to classify any random URL from the smartphone	-Logistic Regression -Naive Bayes -Decision Forest	99.8%
[12]	-Custom	-On the basis of these factors a score will be provided and if it comes out to be less than a certain no than that URL will be considered to be a phishing URL.	99.1%
[13]	-Custom	-Support Vector Machine (Polynomial Kernel) -K Nearest Neighbor -Support Vector Machine(Gaussian Kernel) -Naïve-Bayes -Support Vector Machine (Linear Kernel)	98.60% 98.60% 99.16% 98.88% 98.60%
[14]	- [29], [30]	-Artificial Neuro Fuzzy Inference System (ANFIS) -SVM -KNN	**Text Features:** 98.55%, 94.3%, 95.5% **Frame Features:** 98.06%, 59.99%, 59.59% **Image Features:** 97.20%, 63.30%, 59.20% **Hybrid Features:** 98.30%, 95.20%, 96.10%
[15]	-Feature Selection	-Boosted Decision Tree -SVM -Decision Tree, -Naive Bayes	58.28% 94.74% 93.52% 96.14%
[16]	-450,000-website open-source labeled dataset from Kaggle repository	-Naive Bayes (NB), -K-Nearest Neighbours (KNN), -Stochastic Gradient Descent (SGD), -Support Vector Machine (SVM), -Logistic Regression (LR) -Decision Tree (DT) -Random Forest (RF),	Best Accuracy - Random Forest
[17]	-GitHub URL Dataset	-Multilayer Perceptron Model -Random Forest Classifier Model -Decision Tree Classifier Model -Logistic Regression Model	94.5% 95.2% 96.8% 83.5%
[18]	-Web crawlers	-Malicious URL Acquirement Module, Topic Analyzer, Web-page Analyzer, Comprehensive Analysing and Labelling Module, Attacks Classifying Module and Output.	99.81%
[19]	- host-file.net - phishtank.com - top rankings of Alexa	- Feature Analysis - Feature Semantics - Feature Fingerprinting feature - Feature Fusion	99.89%

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

[20]	- Phishtank - URLhaus - Alexa - Malicious_n_Non-Malicious - Lexical Feature Analysis - Host based feature analysis - Content based feature analysis	- SVM - RF	90.70% 96.28
[21]	-Phishtank -Openphish -Alexa	- Capsule Based Neural Network - 1 Convolution Layer and 2 Capsule Layer - The output of all 4 layers is averaged out to improve the generalization of the approach	99.66%
[22]	-Labelled Dataset with malicious and non-malicious datasets	- Logistics Regression is used - Representation In the form Of 0 or 1. - Dataset contains different features on the basis of algorithm decides whether the url is legitimate or not	98.42
[23]	-HTTP CSIC2010 dataset	-Neural network system	99.14%
[24]	-Alexa, DMOZ, PhishTank, PageRank, WHOIS information	-Naive Bayes -J48 -IBK -SVM	68.60 93.20 88.30 83.93
[25]	-Webpage code feature -Webpage text feature -Quick classification result of CNN-LSTM into multidimensional feature	-Deep learning model	98.99

III. COMPARATIVE ANALYSIS

In this section a comparison between the accuracy of various machine learning models and deep learning models is being drawn.

The Uniform Resource Locator (URL) features can be used to differentiate between a legitimate and phishing website. The URL of a phishing website may be very similar to real websites to the human eye, but they are different in their IP address.

- Domain name portion is constrained since it has to be registered with a domain name Registrar.
- Subdomain name and Path are fully controllable by the phisher.

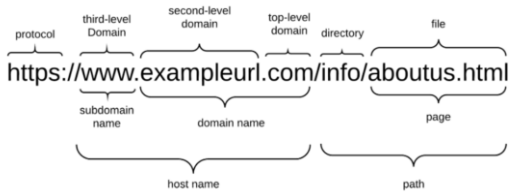

Fig.3.1 URL Features

Dataset [27] includes features like length of url, length of hostname, number of hyphens, whois registration and more to decide if the url is legitimate or phishing.

The performance of ML & DL models on the datasets for phishing websites varies extensively. The ML models seem to have a stronger hold on the numeric data while DL models struggle to reach the optimal accuracy.

Random forest, Decision Tree, Logistic Regression, K Neighbours, XGBoost, XBNet, MLP (using PyTorch Neural Net Model, Churn Model), MLP Model, SVM, Ada-Boost were the models that were chosen. These models were used on dataset(1) [26], dataset(2) [27] & dataset(3) [28].

Fig.3.2 shows the correlation graph of numerical features of this dataset.

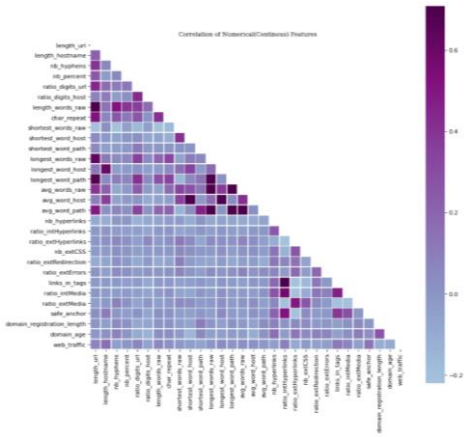

Fig.3.2 Correlation of Numerical(Continuous) Features

Random Forest model showed the best results. The dataset used was from Kaggle [28].

Following *Fig- 3.3* shows the confusion, precision, recall matrix for Random Forest.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

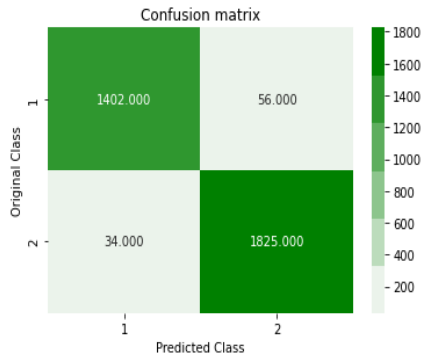

Fig.3.3 Random Forest [confusion, precision, recall matrix]

Decision Tree was one of the most used models in previous studies by different authors. It showed fairly good results and an accuracy over 90%. The dataset we used for DTree was from Kaggle [28].

Confusion, precision, recall matrix for DTree has been shown below in *Fig 3.4*:

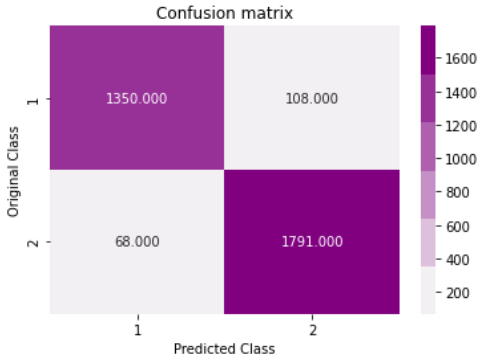

Fig.3.4 Decision tree [confusion, precision, recall matrix]

Neural Network models were rarely experimented with for detecting phishing urls. The accuracy was comparatively lower with average performance and increased train and run time. We experimented with Extremely Boosted Neural Network (XBNet) and used dataset(1) [26] to achieve an accuracy above 50.

XBNet performance is shown in following *Fig- 3.5*:

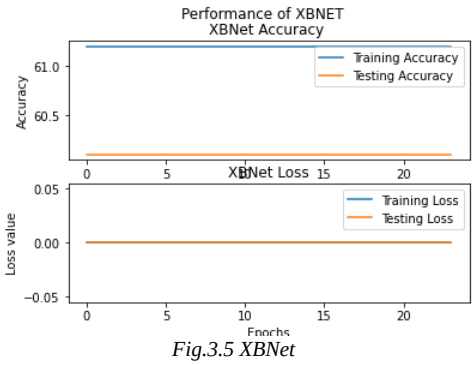

Fig.3.5 XBNet

The model, dataset, accuracy, recall and F1 score are shown concisely in the below *Table- 3.1* and their respective accuracy has been represented in *Fig- 3.6* in a graphical format for better understanding:

Table 3.1 Model Comparison [non : non-malicious urls, mal. : malicious urls]

Model	Dataset	Accuracy	Recall (non)	Recall (mal.)	F1 score (non)	F1 score (mal.)
Random forest	(3)	97.3	0.98	0.97	0.97	0.98
Decision Tree	(3)	94.7	0.95	0.94	0.94	0.95
Logistic Regression	(3)	93	0.91	0.95	0.92	0.94
KNN	(3)	68	0.64	0.71	0.64	0.71
XGBoost	(3)	93	0.64	0.71	0.64	0.71
XBNet	(1)	60.1	0.87	0.35	0.69	0.47
MLP (using PyTorch Neural Net)	(2)	94	0.92	0.92	0.94	0.94
MLP	(3)	70.6	0.97	0.49	0.75	0.65
Ada-Boost	(3)	95.5	1.00	0.88	1.00	0.93
SVM	(3)	80.7	1.00	0.00	0.89	0.00

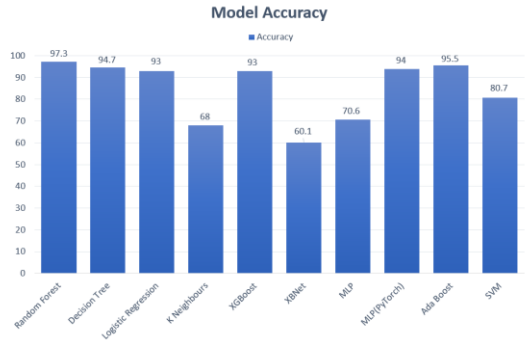

Fig.3.6 Model Accuracy Graph

IV. SYSTEM DESIGN

From the comparative analysis we can conclude that Random Forest was the best performing model and is selected for the tool. The dataset used for training and testing purposes is dataset(2) [27]. The tool is programmed in Python language and is a simple command line tool.

The user is asked to provide the domain and URL for which he wants to check the legitimacy. The input is then sent to two functions, one where it gets vectorized and a prediction is made by the model stating if the URl is legitimate or malicious & second where it is sent for reconnaissance to provide information about open/closed ports, directories and subdomains associated with the domain. A combined report of the outputs is displayed to the user in the end.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

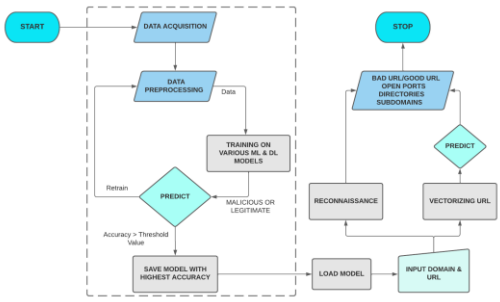

Fig.4.1 Tool Flowchart

Understanding the reconnaissance performed, the port scanner works on the input target domain and checks the 65,665 ports for any open port and returns them to the user. On encounter with any closed ports the flow returns back to scanning. The process is completed after all these ports are scanned. The Directory Brute Forcing tool works on input URL and domain and then checks with every word in the available wordlist for existing directories. It then returns the found directories to the user. The Subdomain Enumeration tool works on input URL/domain and makes a request to crt.sh for existing subdomains and outputs them to the user in json format.

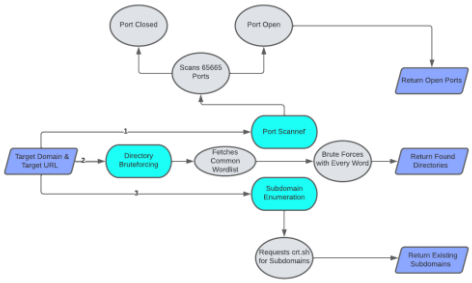

Fig.4.2 Reconnaissance Flowchart

V. METHODOLOGY

Data Acquisition:

A dataset with text URLs labeled as good and bad and evenly divided proportion of good-bad urls is prefered. The data was sanitized by removing NaN valued attributes and redundant data.

	url	target
0	http://www.crestonwood.com/router.php	1
1	http://shadetreetechnology.com/V4/validation/a...	0
2	https://support-appleId.com.secureupdate.duila...	0
3	http://rgipt.ac.in	1
4	http://www.iracing.com/tracks/gateway-motorspo...	1

Fig.5.1 Dataset attributes

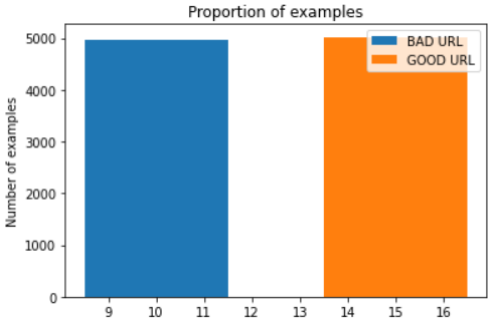

Fig.5.2 Dataset proportion

Cleaning:
As the url length varies significantly among the entries of the dataset, this may add to the bias while we're training and predicting. Hence, we can't rely on word count. The problem is avoided through the concepts of NLP. The textual data is converted to numerical vectors as algorithms are more precise with numeric data. We use Bag of words, Term Frequency, Inverse Document Frequency etc.

However, we don't completely adopt the absolute step-sequence of NLP such as omitting punctuations, stop words, data lemmatization etc as attackers usually make small modifications to make the urls look legitimate.

Model Selection and Training:

Random forest was selected as per the result of our analysis of different models. The model was trained and saved. We also had to save the vectorizer for converting the urls to numeric vectors for predicting the url in our tool.

Integrating the Model to Tool:

The tool does url reconnaissance and detects whether the url is legitimate or not. For predicting the legitimacy the random forest is used.

The tool basically performs 3 different types of scans/reconnaissance.

- Port Scanning
- Directory Reconnaissance
- Subdomain reconnaissance

In directory reconnaissance, the domain and a wordlist is feeded to the tool and then the tool brute forces all the paths present in the wordlist on the domain . After that in result the tool shows out the actual directories present in the domain

Port scanning is the procedure of scanning the network ports of the server to check whether any port is open or not. A domain is feeded to the tool and then the tool resolves that domain name to the corresponding IP Address. After that , A SYN packet is sent to the server port , if the port responds back with a SYN/ACK then that port is shown to be open and if the port does not respond back SYN/ACK then that particular port is shown to be closed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

In subdomain Enumeration, the tool requests crt.sh for all the subdomains that are there for the particular domain and then shows back all the subdomains which have valid certificates in results.

In case of a bad url/malicious url the tool displays the prediction made as 'BAD URL' followed by the scanning process which includes ports, directories and subdomains. Mostly in the case of a malicious website, there may be no directories and/or subdomains associated with the URL. This further confirms that the input URL/domain must be malicious and thus the user should be precautious of such websites.

Fig.5.3 Output for Bad URL

For a website which is legitimate, the tool displays the prediction made as 'SAFE URL' followed by a similar scanning process which includes ports, directories and subdomains. Usually in case of safe/legitimate websites, there are directories associated with the URL as well as the domain has its respective subdomains which further help in confirming its legitimacy.

Fig.5.4 Output for Safe URL

VI. CONCLUSION & FUTURE WORK

From the comparison drawn between all the stated models, it can be observed that Random Forest can most accurately predict the malicious and non malicious URLs. Hence, Random Forest was integrated into a tool to help detect the malicious URLs. As we cannot completely rely on Machine predictions yet, the other details about the URL such as open ports, subdomains etc are also displayed to the target user (cyber security personnels) so that legitimacy of the url can also be judged/verified by the user.

In future, DL models such as MLP churn model or some other high performing DL models can be used that surpasses the accuracy of ML models like Random Forest. Currently the tool is CLI based but it can be enhanced by introducing GUI. Also, shall take care of eradicating or reducing the possibilities of false positive predictions. Improved dataset with enhanced tokenization/vectorization can improve predictions.

ACKNOWLEDGEMENT

We would like to express our gratitude to everyone who gave us the chance to finish this report. Firstly, we want to take this opportunity to express our significant appreciation to the School of Computer Science & Engineering (SET), Sharda University for the support and environment provided to implement it as our major project. Special thanks to our supervisor, Mrs Neha Tyagi and Ms Manpreet Kaur Aiden, whose constant help, support and suggestions helped us in the fabrication process and in completing this research.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

REFERENCES

[1] Suyeon Yoo and Sehun Kim, Two phase malicious web page detection scheme using misuse and anomaly detection, International Journal of Reliable Information and Assurance, Vol.2, No.1, 2014.
[2] Abubakr Sirageldin, Baharum B. Baharudin, and Low Tang Jung, Malicious Web Page Detection: A Machine Learning Approach, ResearchGate, 2014.
[3] Yao Wang*, Wan-dong Cai and Peng-cheng Wei, A deep learning approach for detecting malicious JavaScript code, Security Comm. Networks 2016; 9:1520–1534, 2016.
[4] J Forensic Sci & Criminal Inves, Malicious Website Detection: A Review, Journal of forensic sciences and criminal investigation ISSN 2476-1311, 2018.
[5] Birhanu Eshete, Adolfo Villafiorita, Komminist Weldemariam, Malicious Website Detection: Effectiveness and Efficiency Issues, Proceedings of the 2011 International Conference on Machine Learning and Cybernetics, Guilin, 10-13 July, 2011.
[6] Hyunsang Choi, Bin B. Zhu, Heejo Lee, Detecting Malicious Web Links and Identifying Their Attack Types, ResearchGate, 2011.
[7] Wen Zhang, Yu-Xin Ding, Yan Tang, Bin Zhao, Malicious Web Page Detection Based On On-Line Learning Algorithms. IEEE Xplore: International Conference on Machine Learning and Cybernetics. (2011).
[8] Ping Yi, Yuxiang Guan, Futai Zou, Yao Yao, Wei Wang and Ting Zhu, Web phishing detection using deep learning framework, Hindawi Wireless Communications and Mobile Computing Volume 2018.
[9] Vahid Shahrivari, Mohammad Mahdi Darabi, Mohammad Izadi, Phishing detection using machine learning techniques, ResearchGate, 2020.
[10] Yu-Chen Chen, Yi-Wei Ma, Jiann-Liang Chen, Intelligent malicious URL detection with feature analysis, IEEE Symposium on Computers and Communications(ISCC), 2020.
[11] Husam Adas, Sachin Shetty, Waled Tayib, Scalable detection of web malware on smartphones, International Conference on Information and Communication Technology Research (ICTRC), 2015.
[12] Nureni Ayofe Azeez, Balikis Bolanle Salaudeen, Sanjay Misra, Robertas Damaševičius, Rytis Maskeliūnas, Identifying phishing attacks in communication networks using url consistency features.
[13] Ryohei Komiya, Incheon Paik. Masayuki Hisada, Classification of Malicious Web Code by Machine Learning , 2011.
[14] M.A. Adebowale, K.T. Lwin, E. Sánchez, M.A. Hossain, Intelligent Web-Phishing Detection and Protection Scheme using integrated Features of Images, Frames and Text, 2018.
[15] Yung-Tsung Hou, Yimeng Chang, Tsuhan Chen, Chi-Sung Laih, Chia-Mei Chen, Malicious web content detection by machine learning, 2009.
[16] Shantanu, Janet B, Joshua Arul Kumar R, Malicious URL Detection, 2021.
[17] Jino S Ganesh, Niranjan Swarup. V, Madhan Kumar.R, Harinisree.A, Machine Learning Based Malicious Website Detection, 2020.
[18] Senhao, Zhiyuan Zhao, Hanbing Yan, Detecting Malicious Websites in Depth through Analyzing Topics and Web-page, ResearchGate, 2018.
[19] JIANTING YUAN ,YIPENG LIU , LONG YU , A NOVEL APPROACH FOR MALICIOUS URL DETECTION BASED ON THE JOINT MODEL, HINDAWI, 2021.
[20] Cho Do Xuan, Hoa Dinh Nguyen, Tisenko Victor Nikolaevich, MALICIOUS URL DETECTION BASED ON MACHINE LEARNING, INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS(IJACSA), 2020.
[21] Yongjie Huang, Jinghui Qin, Wushao Wen, Phishing URL Detection Via Capsule-Based Neural Network, IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification(ASID), 2019.
[22] VANITHA ANANDKUMAR, MALICIOUS-URL DETECTION USING LOGISTIC REGRESSION TECHNIQUE, INTERNATIONAL JOURNAL OF ENGINEERING BUSINESS MANAGEMENT, 2019.
[23] Chaochao Luo, Shen Su, Yanbin Sun, Qingji Tan, Meng Han, Zhihong Tian, A Convolution-Based System for Malicious URLs Detection, Computers, Materials & Continua, 2020.
[24] Joby James, Sandhya L. and Ciza Thomas , Detection of Phishing URLs Using Machine Learning, International Conference on Control Communication and Computing (ICCC), 2013.
[25] Peng Yang, Guangzhen Zhao and Peng Zeng, Phishing Website Detection based on Multidimensional Features driven by Deep Learning, IEEE Access, 2019.
[26] Dataset, https://www.kaggle.com/shashwatwork
[27] Dataset, https://www.kaggle.com/shashwatwork
[28] Dataset, https://www.kaggle.com/eswarchandt
[29] The University of California Irvine, Rami et al. (2015A): Text and Frame Data.
[30] Dataset, University of Huddersfield, Rami et al. (2015B): Image data.