XSecure: Machine Learning Based Phishing Malware Detection

Zainab Fatima [1], Kabeer Ahmed [1], Rehan Mumtaz [1], Saad Akhtar [1] , Muhammad Areeb [1], Haris Aqeel [1]

1. Department of Software Engineering, NED University of Engineering and Technology, Karachi 75270, Pakistan;zainab.ned@cloud.neduet.edu.pk (Z.F.); ahmed4201750@cloud.neduet.edu.pk (K.A.); mumtaz4203271@cloud.neduet.edu.pk (R.M.); saad4205499@cloud.neduet.edu.pk (S.A.); nadeem4201543@cloud.nedet.edu.pk (M.A.); aqeel4203783@cloud.neduet.edu.pk (H.A.)

Abstract: With the advancement in technology, the people's assets are exposed to the public making them vulnerable to the open attacks.Among them, *phishing* is a type of online social engineering attack that aims to steal a person's digital identity by impersonating a legitimate entity. The hacker transmits a signal or a message through an email or a link embedded in that email, chat, social media post, or other channel with a link to a malicious site which captures users' private and sensitive information. We're focusing and focusing our efforts on establishing a system for URL analysis and classification in order to prevent phishing assaults and cybercrime. The needs and obstacles for implementing Malicious URL Detection as a service for real-world cybersecurity applications are highlighted in this document. So, malware website detection is the need of today's internet platform and applications of malware detection should be user-friendly and easily accessible to all users.

Keywords: Malicious URL Detection; feature classification; feature extraction; logistic regression; vectorization; random forest

I. INTRODUCTION

Hackers regularly employ spam and phishing to get users to visit malicious URLs, after which Trojans are installed on the victims' systems or sensitive data is exposed [1]. Harmful URL detection technologies can help users discover and avoid malicious URLs and prevent them from being attacked by malicious URLs. In the past, author give in [41] Blacklist-based methods for detecting phishing websites, are utilized. There are a few significant advantages to this strategy. It's quick, has a low rate of false positives, and is simple to use.

To identify malicious URLs, researchers used a machine learning technique. These methods frequently require manual extraction of the features, and to avoid detection attackers can redesign the features. Developing a more effective malicious URL detection model has become a research priority in today's complicated network environment.

This paper provides a strategy for detecting fraudulent URLs that is based on a logistic regression model using machine learning techniques. This machine Learning security solution, supports an always on detection system.

Our contributions and advances in this study are as follows:

(1) This paper explains how to apply a model for detecting fraudulent URLs that is based on a Logistic Regression Model with better accuracy like 96% than other previous models like SVM(Support Vector Machine) Classifier with an accuracy model of 77% ,AdaBoost Classifier with an accuracy of 80% ,and Random Forest Classifier with an accuracy of 86% etc.

(2) The previous research shows of different ways of utilizing the dataset and using it for their model but as the technology advances it creates endless techniques to bypass the validation in place for a scanner,therefore we proposed some different vectors & techniques to overcome of these bypasses and to give blackhats a hard time in doing their dirty work. We define it in the methodologies section. The features are taken from the URL sequence during the feature extraction and representation step [4]. The identified and extracted characteristics are transferred into a vector, which is then processed immediately by machine learning to learn the classification model. Extracting features like “@” in the URL can redirect you to some other malicious website so if there is an “@” in the URL link the trained model will detect the website as malicious.

(3) caAnother idea we think of is to take the help of the open source or open-public databases/ ranking-websites mechanisms, that identifies the behavior based on the website signature. We demonstrate the practicality of the paradigm described in this study. We find malicious URLs with the embedding approach such as status Alexa index, status abnormal URL, status domain age. status redirects, status prefix suffix, status '@': not applicable. If either of these has a false status then it means the website is not safe for browsing.

(4) Furthermore, in our recent work we successfully implemented our model to an extension prototype for browser that means user don't have to visit our website every time as browser extension would come to rescue to show results on the page visited. All its implementations & results are shown in result section

II. RELATED WORK

As we know that we entered the era of technology a few years ago, the researchers are getting into the cybersecurity domain with more interest. So in the past many authors have written research papers on malicious detection using various approaches. Some of their work is discussed below:

II.I APPROACHES

Signature based Malicious URL

Detection For a long time, researchers have been researching and applying signature [6] sets to detect bad URLs. Lists of known harmful URLs are typically used in the bulk of these investigations. When a new URL is visited, a database query is done [14,15]. If a URL appears in the list of blacklisted URLs, it is harmful and will raise an alert or warning; otherwise, URLs are regarded as safe. The primary problem of this [8,9] technique is that new malicious URLs that are not included in the given list would be exceedingly difficult to detect.

Machine Learning based Malicious URL Detection

supervised learning, unsupervised learning, and semi-supervised learning are three types of machine learning algorithms that may be used to detect harmful URLs [13,16]. The detecting algorithms are based on URL behaviors. Based on machine learning methods, [2,4,5] analyses a variety of harmful URL systems. Machine learning methods such as SVM, Logistic Regression, Naive Bayes, Decision Trees, Random Forest, and others include SVM, Logistic Regression, Naive Bayes, Decision Trees, Random Forest, and others. This article uses the Logistic Regression Model. The experimental results will demonstrate the algorithm's accuracy with various parameter configurations.

URL properties and behaviors may be classified into two categories: static and dynamic. In their works [12, 13, 19], authors presented Lexical, Content, Host, and Popularity-based approaches for evaluating and extracting static behavior of URLs. As machine learning algorithms, these experiments employed online learning algorithms and SVM. [21, 22] use dynamic URL operations to identify malicious URLs. Character and semantic groups, as well as the Abnormal group in websites and the Host-based group; the Correlated group [24], are all explored [8].

II.II Detection TOOLS

According to the investigation and assessment of harmful URL detection tools described below, the majority of existing malicious URL detection programs [16] are signature-based URL detection systems.

Tools	**Working**	**Performance**
URL Void	Cross-referencing a website with more than 30 blacklist engines and online website reputation services is how URLVoid assesses a website.	This tool uses different search engines across the internet to produce results of safe/phishing url with the

		autonomous system number
UnMask Parasites	Unmask Parasites is a straightforward internet security solution that aids in the discovery of _hidden illicit material (parasites) that cybercriminals smuggle into legitimate web pages utilising a variety of security flaws.	This tool has the benefit of rapidly and reliably detecting iframes.
Comodo Site Inspector	The account owner is promptly alerted via email if malware is identified or if the website is found on any of a number of website blacklisting services.	No longer will businesses have to wait for angry customers to complain that their website contains malicious content. To take advantage of this essential service, webmasters just need to take a few minutes to sign up and configure the service. SiteInspector will do the rest.

Paper Title	Technique/Methodology Proposed	The Issue Highlighted / Addressed	Proposed Architecture: Positive and Negative Points	Network Security Strong/Weak Also Named Security Schemes Applied	Technology
[26]	This study provides a natural language processing-based approach for detecting malicious URLs. On blacklist words, we employ features such as word vector representation derived using Global Vector for Word Representation (GloVe), statistical cues, and n-gram.	Maliciousness of URLs is one of the most common techniques employed by attackers to perform targeted phishing attacks on the Internet	ANN architecture is proposed. The paper highlights the benefits of using Artificial Neural Network in today's research machine learning projects.	This study does not propose or use any network security scheme.	Global Vector for Representation (Glove statistical feature
[27]	Support Algorithmic Vector Machine is utilized in this study to achieve autonomous learning and build the classifier; classification through TF-IDF is used to process the data, and after normalization and standardization, the feature matrix of the gathered URL data is obtained and saved in the sparse matrix	As individuals become more Internet reliant, concerns regarding security of the network have grown in prominence and increment of bad malicious websites have also emerged. To achieve proactive and efficient malicious web page detection interests' students study priority in the field scope of network security all around the world.	The system clusters and reduces the dimensionality of the retrieved features using the K-Means and TSVD algorithms. In terms of sparse matrix processing, the K-Means approach outperforms the TSVD technique, moreover excessive data dimensions raise the model's complexity. .	There is no sign of any security frameworks for this system, therefore it can accept the flaw.	SVM detection model
[28]	This paper proposes the use of a malware identification method that examines the URLs browsed by a programmable automation bot to detect malware. .	Malicious software's scope and diversity have increased dramatically on mobile networks, posing a substantial threat to users' property and personal privacy.	In addition, we do extensive tests to compare the suggested method to others and confirm the detection model's validity. The results of the tests reveal that our approach detects malware reliably and quickly. Not only it can detect malware detected over the course of a year, but also helps in detect potentially malicious programs in 3rd-party application stores.	There are no particular security precautions mentioned in the document.	multi-view neural netw

[29]	The paper investigates the darknet's URL addresses and presents an artificial neural network-based solution for darknet URL recognition.	Due to the increasing criminal activities, mostly takes help of darknet which makes both parties as anonymous protects relationship between two sides of communication from being leaked. Because the IP addresses of both participants to the connection cannot be identified on darknet, the user's identity could not be exposed.	The paper employs an artificial neural network to detect darknet URLs by extracting a numeric vector from the URL. URL length numeric vector can be changed on the fly, and the URL content can be restored in part. Varied lengths of numeric vector of URL can have different categorization effects at the same time.	There are no particular network recommendations for analyzing security restrictions in this document.	Artificial Neural netw
[30]	Paper adopts machine-learning method and uses passive DNS as the analytical data to construct a malicious domain name classification detection model.	Domains with malicious intent is dangerous, the area of threats with the scope it possesses are continuously growing, with the use of traditional reputation systems and reverse engineering methods to detect malicious domain name cannot be real-time, and the process of detecting malicious domain name is complicated and cumbersome.	For the detection of pornographic domain name, we propose a scheme to convert the access characteristics of a domain name into a word vector using the Word2ec model. This solution is completely different from the previous detection schemes, which based on user access logs or web content	N/A	word vector tool Word
[31]	This paper proposes a model to develop a method that can detect phishing attacks by just using only nine lexical features. Various machine learning classifiers were tried, with the Random Forest (RF) classifier achieving the greatest accuracy of 99.57 percent.	Fake websites hosted by attackers seems very appealing and authentic to the users. When user enters their credentials, these are redirected to the attackers. Sometimes hackers attack the user's device and inject the malware in it and change to the botnet. Attackers use these botnets to execute DDoS and several other attacks.	Anti-phishing system that can detect phishing URLs in real-time without the need for any third-party data and with a very fast response time. The proposed approach is incapable of detecting harmful material contained in small short URLs.	No such network security scheme is proposed or used in this paper.	Feature extr algorithms
[32]	The proposed methodology analyses lexical features such as URL and domain length, URL encoding, number of	Blacklists may be in the form of IP addresses or websites used by email filters and block the users through an available list of IP	The webpage's rank is determined by host-based factors such as page rank. Phishing web pages have a limited lifespan and a less age domain. As a	There is no such network security technique presented or employed in this article.	Host based and l based features with classifier adapted in filters

	special characters or malicious IP addresses as well as host-based features like page rank and domain lifespan, to find maliciousness or beingness of the email	addresses or websites. One of the biggest issues with blacklists is that they don't detect phishing URLs in real time.	result, they have an extremely low page rank or none at all.		
[33]	The approaches used in this work were a mix of both linear and non-linear spatial transformations. For linear, a two-stage distance metric was designed. The Nystrom technique was employed for kernel approximation in nonlinear transformations, with updated distance metric.	In real time, blacklisting tactics are unable to detect a website's maliciousness. On the other side, content-based classification tries to detect potentially hazardous websites by analyzing the contents or layouts of the pages. Using IP looking-up programming tools, fraudulent sites can recognize the IPs of search engine crawlers and auditing organizations, and then offer alternate content to the auditing institutions and the general public. To different search engines the contents of the fake sites would appear normal and the malicious content can't be detected	This research develops new features that untangle the linear and non-linear relationships between characteristics in hazardous URLs data using five space transformation models (singular value decomposition, distance metric learning, Nystrom techniques, DML-NYS, and NYS-DML). Linear correlations between features and manually picked attributes may not be important to the classification job because the importance/weight of each feature, as well as the scales of the features, are unknown.	Network Security strong and Variational Auto-Encoder (VAE) was introduced as a representation learning method for network security tasks.	Linear as well as Non-space transform methods
[34]	This study proposes a light-weighted technique that simply considers the URL's lexical properties. The results reveal that the Random Forest (RF) classifier is more accurate than the other classifiers.	Assigning weightage to the rules and determining a threshold value for each rule are issues with the heuristic rule-based detection technique. The threshold value may vary depending on the dataset.	Only from the URL, a minimum set of ideal features is extracted, which decreases execution time and data consumption. There is no classifier that outperforms other classifiers in terms of efficiency, accuracy or execution time.	There is no such network security technique presented or employed in this paper.	Logistic Regression Nave Bayes (NB), S Vector Classification (Random Forest (RF K-Nearest Neig (k-NN))
[35]	This research offers a unique malicious domain identification model based on unsupervised learning, and	Neural networks were used in context of supervised learning, which necessitated a vast amount of training data as well as	The accuracy and F1 score of the architecture model remained near 0.99. With only a modest quantity of training data, the model can frame out common	In this paper, no such network security scheme is proposed or used.	CNN and autoencoder designed linear neurons

	builds a domain detection which is malicious approach employing the autoencoder solely using benign domains provided during model training, resulting in a 99 percent detection accuracy and F1 score.	significant data gathering and labelling burdens.	patterns of benign domains. The detection model is unable to determine the malicious domain's sources.		
[36]	They use a variety of ways to thoroughly examine the models' performance. After the algorithms were changed and using the F1 score to measure performance, the result increases considerably from 0.14 to around 0.90. Using the ultimate well-trained model, we can correctly predict the safety of any URLs on the Internet, protecting our personal information and data.	Not all URLs are completely safe. We must exercise caution while clicking on those URLs, as you may unintentionally enter some hazardous URLs. With the recent outbreak of COVID-19, an increasing number of people have been forced to stay at home. To keep up with what's going on outside, the majority of them utilize their technical devices to access the Internet.	As Internet traffic rose during the COVID-19 pandemic, hackers exploited malicious URLs to attack Internet users and steal their information. I use machine learning to develop a high-performing prediction model to discover them.	N/A	Fine Tuned Classifi Model
[37]	To detect hazardous URLs, this study recommends Using a Convolutional Neural Network with Multiple Layers (CNN). One layer of CNN was studied initially in the suggested model. Following that, a two-layer CNN will be used to boost accuracy. The accuracy of detecting bogus websites rises from 89 percent to 91 percent when two layers of CNN are added in the algorithm.	Most of the time, people trust the website and input personal information without considering if it is genuine or not. A malicious website can be used by cybercriminals to launch ransomware attacks, steal passwords, and commit financial fraud. At the Centre of these cybercriminal operations is the social engineering-based Phishing attack.	CCN architecture is proposed. The proposed detection method was initially put to the test on one CNN layer. After that, two layers of CNN are used to increase the detection accuracy. The model's accuracy as well as performance were evaluated, and positive results were revealed.	N/A	Multilayer CNN

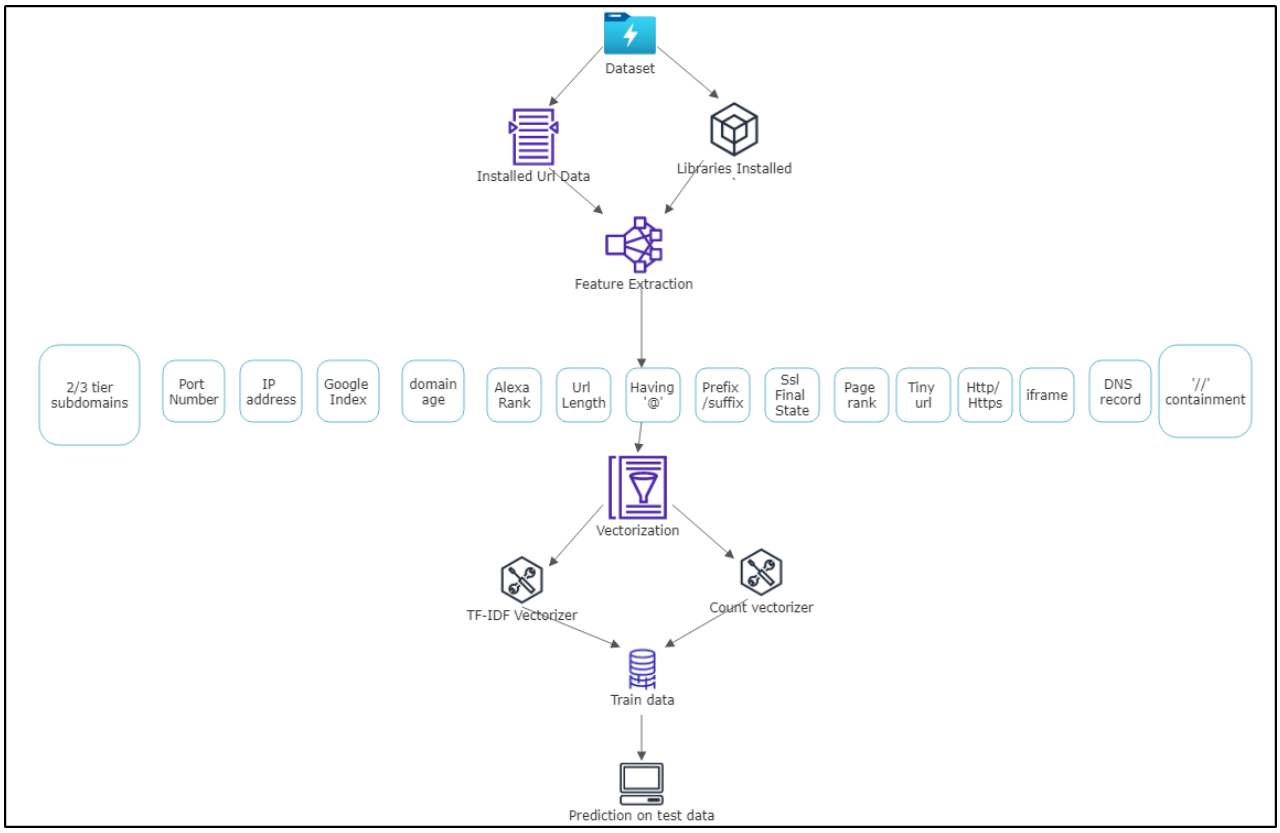
[38]	The proposed technique proved successful in exposing the patterns of different types of fraudulent domain names in groupings It outperformed previous methods. and gave a clearer view of the data's common patterns when used to evaluate a blacklisted collection of URLs in a genuine business network.	Monitoring the URL list in network traffic data might reveal anomalous behavior. Malware is commonly transmitted through email and rogue websites.	It dynamically visualizes the dynamism of the attack pattern. It contributes to a damaging website ecology by reducing the number of black listings. It leverages open-source threat information to validate the risk level and damaging category.	N/A	Blacklist and mal URL extension with F
[39]	The author of this paper has created a strong foundation for swiftly and automatically detecting phishing URLs. They tested their method on an actual dataset and were able to reach an accuracy of 87 percent in real-time.	A phishing attack impersonates a trustworthy third party in order to get sensitive information from a victim. In such an attack, users are typically routed to a fake website that appears to be legitimate. The URL of the phishing website is frequently communicated by email or instant messaging.	They've laid down a solid foundation for detecting phishing URLs automatically. To deal with the limitless increase of URL space, they employed online learning.	N/A	Lexical, Host I Domain WHOIS Base GeoIP Based Features
[40]	The research suggests employing to discover harmful Uniform Resource Locators, researchers used a convolution neural network and a Recurrent Neural Network with Extended Present Moment Memory as models. A recurrent neural network with extended short-term memory achieved the highest accuracy of roughly 98 percent for categorizing phishing Uniform Resources.	Spam mail & phishing websites are widely used to aid certain assaults, which trick the client into revealing credentials and other sensitive information. Email is frequently thought to be the major mechanism of propagating a wide range of malicious assaults.	Deep learning methods such as RNN and RNN-LSTM are preferred to AI approaches since they may give outstanding component portrayal utilizing only raw URLs as data.	N/A	RNN and RNN-LSTM models

Paper Title	Strong Point	Weak Point	Future Work for Scope
[26]	A proper architecture is proposed with strong research simulated neural network with Global Vector for Word Representation (GloVe) as one of the elements that boost machine learning's accuracy in detecting dangerous URLs	Does not discuss the future improvements. Does not mention the challenges faced in implementing on the system.	The different architectures that improve the efficiency a performance of system even further.
[28]	Our method is ideal for situations when the only features provided are URLs. Some web servers, for example, just keep URL data and all other data in network transmission is rejected.	This model cannot predict/identify the maliciousness in the URL using some protocol other than HTTP in the traffic	Encrypted communication over the traffic and oth protocols, we're considering creating a solution identifying malicious traffic over the network utilizing oth protocol other than HTTP which gives advantage of usi API of VPN to give us full access over their network traff Before transferring traffic for analysis purposes maliciousness behavior detection, a gateway forward forwards the packets in a secure manner via VPN interfa with payload data on the traditional socket interface. By t method, one can deal with the challenge of malw detection with ambiguous fully traffic.
[27]	It uses the TF-IDF algorithm to analyze and optimize the text properties of URL addresses. When given a good classification effect, TF-IDF feature extraction can describe each morpheme of the feature in depth, and the SVM model displays better accuracy with the increments in the dataset	This model offers complex implementation that required a broader scope to work with at the same time needs larger datasets to produce greater efficiency of results, with smaller datasets efficiency gets smaller and smaller	There are still vector features that can be added to featu vector selection, and additional techniques to be featur extracted. The new ones can be added in the future, a alternative feature extraction methods can be tried improve the identification and the system's detection rate.
[29]	This research employs an artificial neural network to detect darknet URLs by extracting a numeric vector from them. The adjustments in URL's numeric vector length and extracting partially content from the URL. Different lengths of URL numeric vectors can have varying classification impacts at the same time. .	Model do not recognize spamming, phishing, & malicious URLs in darknet	The article only uses an ANN model and does not look in the effects of other network models like CNN, RNN, lo short-term memory, or transform models on detect outcomes. These studies will be carried out in the future.

[30]	This solution is completely different from the previous detection schemes, which based on user access logs or web content. We organize the domain name records in the passive DNS by time and user IP, and then use the deep learning mechanism to embed the domain name into the vector space	The Model Accuracy is not quite good enough	There's still a lot of space for growth in the attenuation eff of the time dimension. Therefore, we can consi combining multiple classification modes or to use popular LSTM model in deep learning to classify
[31]	The method employs only limited number of features based on the lexical aspects of URLs, resulting in a 99.57 percent accuracy and reduced execution time and dataset storage needs.	Lexically different URLs from the nine features used cannot be detected as malicious. Haven't discuss in detail what future work will be done and by using which methods and technology	Use of more sophisticated based lexical features of UR and using more advanced algorithms.
[32]	Use of lexical features as well as host-based features like page rank and domain age.	Haven't discuss what work will be done in the future and what additional methods can be used.	NA
[33]	Rather than creating new classifiers, focuses on improving the identification of dangerous URLs using feature engineering.	Testing time cost more means slow in predicting the labels of the instances and more training time is required.	Based on the features of the URLs, classifiers can improved achieving less training time and higher accura To investigate unique features of URLs embedded graphical QR codes and to eliminate QR code fra detection.
[34]	This study provides a methodology for detecting and analyzing malicious URLs with a smaller number of attributes retrieved solely from the URL. As a result, execution time and storage requirements are reduced.	The Random Forest delivers the highest accuracy compared to the other classifiers, although KNN performs better in terms of execution.	The research will be expanded in the future using new feature datasets and the application of reduction techniq to minimize execution time and improve accuracy.
[35]	Despite being trained with a tiny training set, the model maintains its detection performance. As a result, the model reduces the time spent accumulating training datasets.	The detection model is unable to determine the sources of malicious domains.	Future research could lead to a better feature extraction a representation that can describe the origins of malici sites.
[36]	They use variety of characteristics to train and evaluate the model as well as optimize hyperparameters. The F1 score is close to 0.92, and the model's accuracy is 97 percent. To aid consumers in spotting bogus URLs, the concept	N/A	The many characteristics that boost the system's efficien and performance even further.

	might be implemented as software or browser extensions.		
[37]	The multilayer CNN will extract relevant patterns in the data from given URLs through using several convolutions with different kernel sizes. The model is more efficient and independent of feature engineering because to the self-extraction process in the design of a multistage trainable neural network	To achieve a better model for the job, the CNN model The number of layers, the number of kernels, the size of the kernels and the optimizer may all be changed.	It may be compared against standard to see if the automa feature extraction-based deep learning model is applicab use machine learning-based classification models l Random Forest, Naive Bayes, Support Vector Machi Logistic Regression, and others.
[38]	Even if antivirus programs do not identify it, every unfamiliar website or URL that is grouped in a cluster with harmful websites should be explored further.	On the basis of the obtained groupings, no clustering or community analysis was attempted. Other security visualization techniques to properly portray distinct harmful patterns were not used.	It should design and implement a group aggregat technique that is buttoned up.
[39]	They used selective sampling and delayed feature capture to increase the system's performance. Their program can find URLs that have never been seen before with an accuracy of 87 percent.	They didn't include n-grams, DNS query results, web page network traffic, bag of words, black list presence, web page content, and other features.	It should add time-varying URL characteristics in the futur
[40]	They are able to provide assurance. Based on their findings, they think that vindictive URL recognition powered on AI and deep learning can replace boycotting and traditional articulation approaches in discovery frameworks.	The extraction of web page detail & content was not included.	It should turn their technique into a module for use in a w application

III. METHODOLOGY



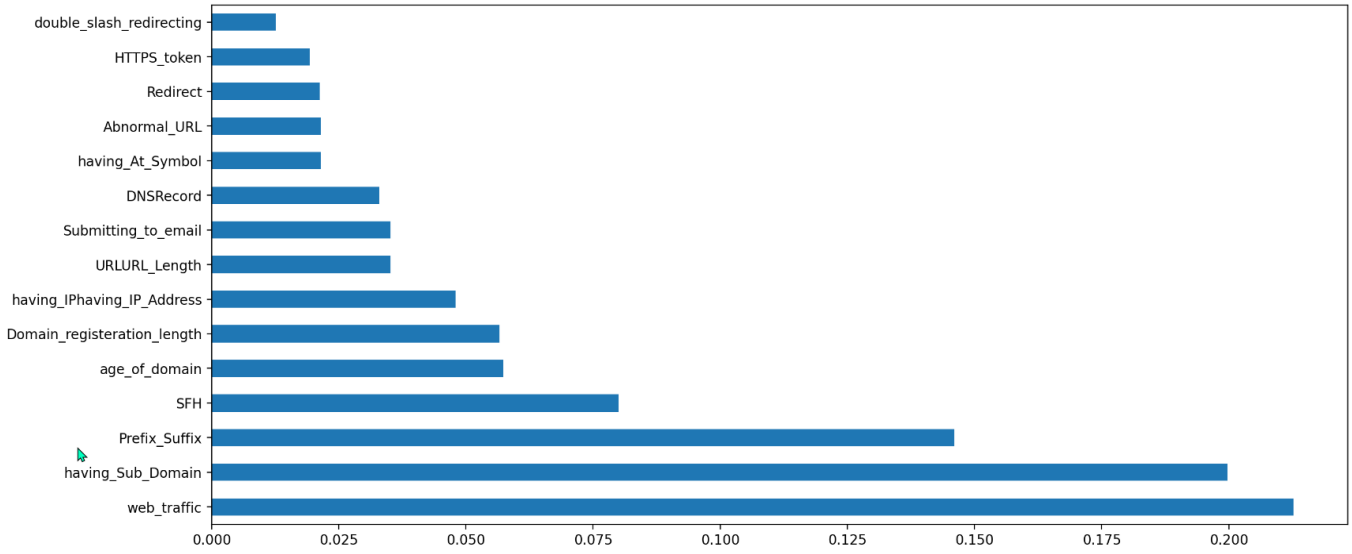
We initiate with our hand getting dirty on the dataset. The dataset acquired is not much in a state to be used in our algorithm, so the first thing we do is to clean our dataset. Extracting the features is another strategy to minutely analyze the data as it can increase precision in our accessing/ predicting the result. Multiple factors are correlated in this whole phase so it has to undergo a certain vectorization phase to vectorize the data. There are built in vectorizers out there as there is always room for customization as it can be useful for increasing the efficiency of the model. In the vectorization phase, we built a specified vector based upon the url after going under the feature extraction. The features if found marked as 1 and if not marked as -1. This data is treated as both testing and training model by splitting into two halves for each sophistication analysis Mainly talking, there are two steps to this model: training and detection.

a) Extraction Phase & Training stage:

For detection of malicious URLs, both malicious and legitimate URLs must be collected. From Kaggle, we obtained a large dataset constituting around more than 4,00,000+ URLs. We gathered around this large dataset of training the model as powerful so that it can give correct results as possible.

Now here comes the feature Extraction step, where the URL is further broken down into the matching 11 listed below protocols and then maintain a vector array to which the match is there If it matches the tokens, it will list the certain token as -1 or otherwise as 1. After those 11 protocols columns there is another column of Result which tell -1 for Malware and 1 for Legitimate URL

These composed vectors are dumped in CSV format and then it get feeded to the training phase as an input for the algorithm. The main comprised tokens are



Factors

1. URL Length: Lengthy URL makes it ambiguous for the user to check, the user only checks for the main domain and let it go the other path

2. Contain @ Symbols: Most of the time, it happens like '@' is there in the URL and no one notices as the browser normally executes the link after '@' symbol

3. Two and Three-Tier Subdomains: Normally URL do not have more than 3 domains under the main name embedded in the URL

4. Suffix & Prefix Separated Domain: Legitimate URLs rarely contain the dash (-) sign. To make a phishing website look like a legitimate website, prefixes or suffixes separated by (-) might be added**.**

5. IP address in the URL: To deceive users and steal important information, an IP address can be substituted for the domain name.

6. ‘//’ makes Redirection: It is possible to determine whether there is any unneeded redirection to other websites by looking at the placement of the '//' in the URL.

7. URL Shortening: Shortening of "URL" refers to a technology in use on the "World Wide Web." that allows a URL to be significantly reduced in length while still directing to the desired web page.

8. Https Webapp: This token can be used by attackers to deceive users.

9. Traffic Control Web: The popularity or ranking of a website influence whether it is a phishing website or not.

10. Domain Life Period: Phishing websites usually only exist for a short time. Some respectable websites are active for at least 6 months, while the majority are active for at least a year.

11. Database Records: Check whether the URL is listed in the phishing URL lists supplied by Phish Tank and others.

b) Detection phase:

Following feature extraction, the algorithm receives the processed data set as input, and their results are compared.

After Successful training of the system, the later part which has been left off that is 10% is used to test the accuracy and precision of the model’s algorithm

The input to the machine learning algorithms comes from CSV files for both training and testing, with training accounting for 90% of the total URLs in the data set and testing accounting for 10%.

The algorithms used are random forest and logistic regression.
Each input URL is subjected to the detection phase. The attribute extraction method will be performed first on the URL. These characteristics are then given into the classifier, which decides if the URL is legitimate or malicious.

IV. PERFORMANCE ANALYSIS

We have tested this application for suspected URLs published on phish tank and tailed results with Google’s Safe Browsing technology.

1. Alexa Ranking:

With Alexa ranking we are getting the ranking of that specific URL whenever it is searched:
Lower the rank more the probability of being legitimate.
Higher the rank the more the probability of being Malicious.

2. Domain registration:

We have marked URLs with registration time of more than 6 months as safe. URLs must qualify for at least one category to be safe either it should have domain registration with more than 6 months registration or it should have lower Alexa ranking.

3. Dataset of Malicious and Non-Malicious URL :

In [17] 400,000+ URLs tagged data source The safe URLs account for 83 percent of all URLs in our database, whereas the harmful URLs account for 17 percent. Two subsets of the above-mentioned dataset of both safe and malicious URLs have been created. The dataset is utilized for training around 80% of the time, with the remaining 20% being used for testing. For our machine learning approach, we used a number of classifiers, trained the model on those classifiers and obtained different results [23]. Specifically, we have used Logistic Regression (LR) and Random Forest classifier. We then calculated different metrics based on these classifiers like accuracy, recall, precision and f1-score. First let's look at these terminologies;

a. Accuracy

Accuracy is one metric for evaluating classification models. Informally, **accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

b. Recall

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.
What proportion of actual positives was identified correctly? Mathematically, recall is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$

c. Precision

Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV).It defines what proportion of positive identifications was actually correct?Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

d. F1-score

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

e. Confusion Matrix

A clean and unambiguous way to present the prediction results of a classifier is to use a confusion matrix (also called a contingency table).

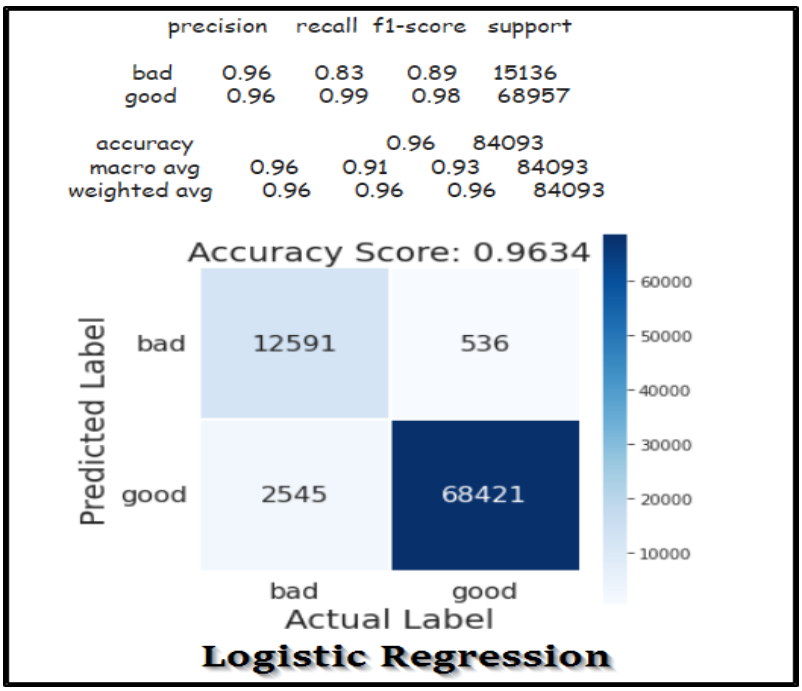
V. Result

We will be using the above terminologies to access the working and efficiency of our Machine learning model. metrics have been calculated by using the following equations.
The detailed results from classifier is shown as

	True Positive	True Negative
Predicted Positive	12591	536
Predicted Negative	2545	68421

Measure	Value	Derivations
Sensitivity	0.8319	TPR = TP / (TP + FN)
Specificity	0.9922	SPC = TN / (FP + TN)
Precision	0.9592	PPV = TP / (TP + FP)
Negative Predictive Value	0.9641	NPV = TN / (TN + FN)
False Positive Rate	0.0078	FPR = FP / (FP + TN)
False Discovery Rate	0.0408	FDR = FP / (FP + TP)
False Negative Rate	0.1681	FNR = FN / (FN + TP)
Accuracy	0.9634	ACC = (TP + TN) / (P + N)
F1 Score	0.8910	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.8723	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Confusion Matrix:



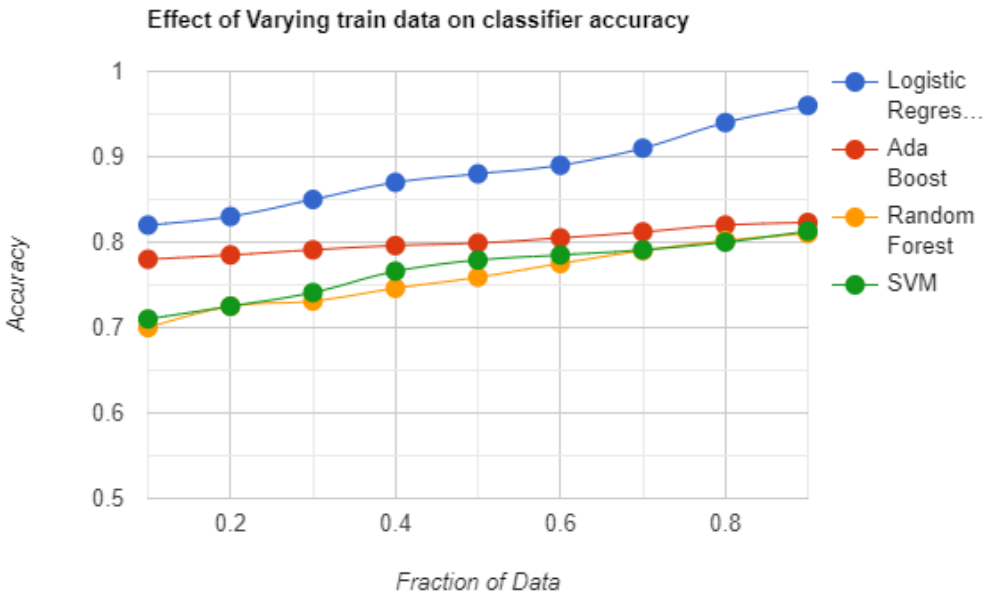
V.I. Comparative Analysis

As can be seen from the below table, the logistic regression model gave the best results. The logistic regression model produced the highest precision score of 96%, highest recall with score of 91% and the highest f1-score of 93.5%.

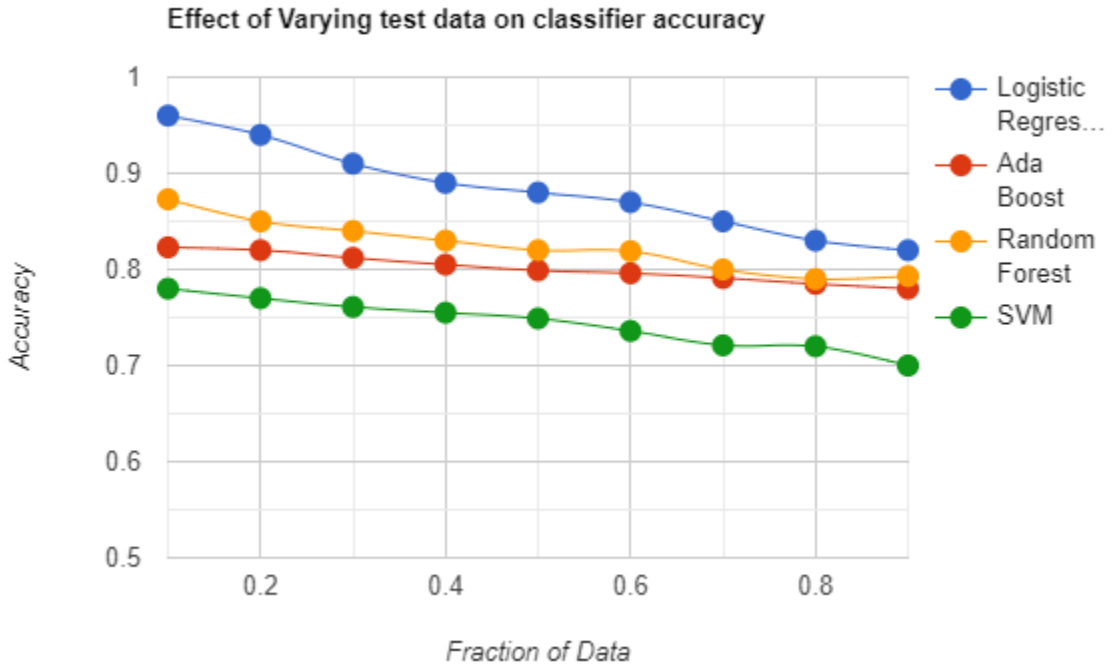
Comparison of both the Classifiers:

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	**96%**	**96%**	**99%**	**98%**
Random Forest	**87%**	**87%**	**90%**	**88%**
ADA Boost Classifier	**89%**	**89%**	**91%**	**90%**
SVM Classifier	**78%**	**80%**	**81%**	**80%**

From the above table , we can find that Logistic Regression is better than the rest in accuracy , precision , recall and F1-score.



This graph illustrates the effect of the fraction of data on the accuracy based on train data, the comparative illustration depicts about the different classifiers behavior on the model accuracy

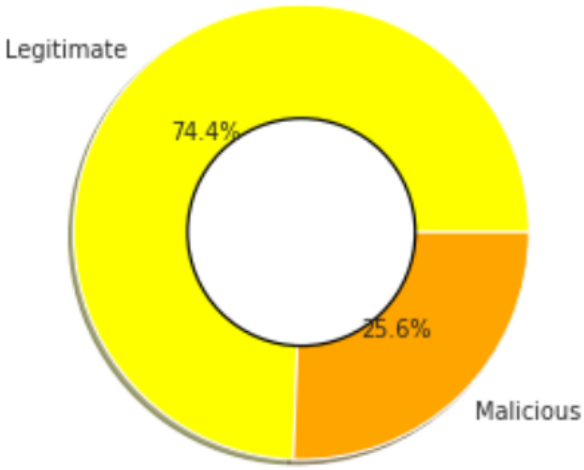


In this graph, the projection is coming downwards as the test dataset is keep on increasing this illustrates the less the data, the more accurate are the results from the machine learning model. The accuracy is keeps on declining if we keep on increasing data fractions

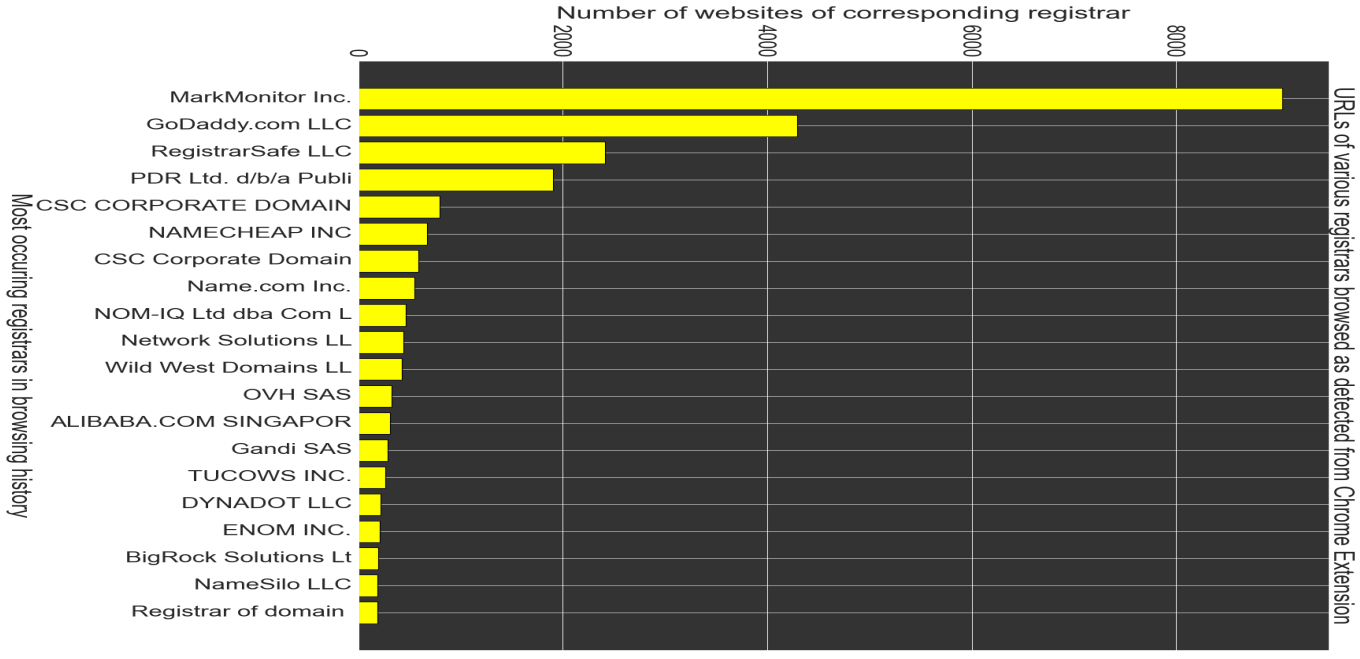
VII. X-Secure Tool UI Analysis

a) Web Interface

The web tool is made for anyone to check the state of the website. Our tool provides a number of services, a one place solution to be said would not be wrong for that. After getting satisfiable accuracy of our Machine learning model, we move towards another iteration of our research to look for better rooms for greater provision of services. Variety of categories can be access, it includes



- *Search* feature lets look for the maliciousness in the website, besides it gives some other rankings too which includes domain information , domain ranking and website characteristics
- *Live Data Analysis* belongs to the extent of work related to chrome extension which will be discussed afterwards. This utilizes our browsing history and store the data results in this page with the illustration through graphs

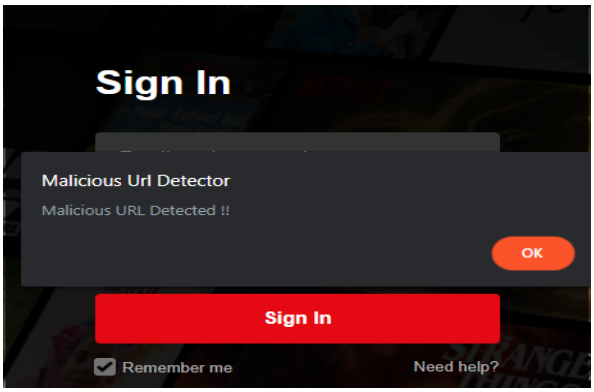


- *Url history* shows the history of url a person has searched through this tool or browses website, in this case it gets picked up by extension

- *Sandbox* feature provides a user to load the website if he thinks it can be harmful for his PC, in that way sandbox loads the website and blocks any malicious links and viruses to be downloaded instantly

b) Chrome Extension

It's a second iteration work product which is much helpful in many ways, like it doesnt need extra configuration. It runs like a web crawler, gives instant result as soon as someone visits website, it analyzes the webapp and gives off result by showing the popup



VIII. CONCLUSION & Future Work

Malicious URL recognition is used in many cybersecurity scenarios, and machine learning methods have a bright future. This article sets out the requirements and challenges for developing Malware Detection Identification as a service for real-world cybersecurity applications. Our solution will use the Chrome Extension to identify harmful links and their origin signatures (first posted person-profile URL, name, email, phone number, etc.) in real time and issue advisory notifications to public corresponding agencies regarding the links' source reliability. Machine Learning is used to evaluate whether a URL is dangerous or not.

We might not be perfect but we have a sharp edge over every other similar application due to highly accurate results, multi-platform support and various other functionalities. This application has the ability that it can be integrated with the other modules which will be created against the cyber-attacks in the upcoming cyber era. This open source has the vision of adding more technologies for malware detection, the upcoming iteration would be focusing for desktop application for system level application behavior detection which will be usable as mostly system users faces virus related issues mostly so mechanism should be made for everyone's safety and smooth work

REFERENCES

[1] D. Sahoo, C. Liu, S.C.H. Hoi, "Malicious URL Detection using Machine Learning: A Survey". CoRR, abs/1701.07179, 2017.

[2] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[3] B. Eshete, A. Villafiorita, and K. Weldemariam, "Binspect: Holistic analysis and detection of malicious web pages," in Security and Privacy in Communication Networks. Springer, 2013, pp. 149–166.

[4] S. Purkait, “Phishing counter measures and their effectiveness– literature review,” Information Management & Computer Security, vol. 20, no. 5, pp. 382–420, 2012.

[5] Y. Tao, “Suspicious url and device detection by log mining,” Ph.D. dissertation, Applied Sciences: School of Computing Science, 2014.

[6] A. Blum, B. Wardman, T. Solorio, and G. Warner. Lexical feature based phishing url detection using online learning. In AISec, pages 54--60, 2010.

[7] Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. 2013. Cross-layer detection of malicious websites. In Proceedings of the third ACM conference on Data and application security and privacy. ACM.

[8] Sandeep Yadav, Ashwath Kumar Krishna Reddy, AL Reddy, and Supranamaya Ranjan. 2010. Detecting algorithmically generated malicious domain names. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM.

[9] Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In Proceedings of the 25th international conference on Machine learning. ACM.

[10] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, “Detection of malicious web pages using system calls sequences,” in Availability, Reliability, and Security in Information Systems. Springer, 2014, pp. 226–238.

[11] R. Heartfield and G. Loukas, “A taxonomy of attacks and a survey of defense mechanisms for semantic social engineering attacks,” ACM Computing Surveys (CSUR), vol. 48, no. 3, p. 37, 2015.

[12] Internet Security Threat Report (ISTR) 2019–Symantec. https://www.symantec.com/content/dam/sy mantec/docs/reports/istr-24- 2019-en.pdf [Last accessed 10/2019].

[13] Wei Zhang, REN Huan, and Qingshan Jiang. 2016. Application of Feature Engineering for Phishing Detection. IEICE TRANSACTIONS on Information and Systems (2016).

[14] Wei Wang and Kenneth Shirley. 2015. Breaking Bad: Detecting malicious domains using word segmentation. arXiv preprint arXiv:1506.04111 (2015).

[15] Yao Wang, Wan-dong Cai, and Peng-cheng Wei. 2016. A deep learning approach for detecting malicious JavaScript code. Security and Communication Networks (2016).

[16] Weibo Chu, Bin B Zhu, Feng Xue, Xiaohong Guan, and Zhongmin Cai. 2013. Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In Communications (ICC), 2013 IEEE International Conference on. IEEE.

[17] Malicious_n_Non-MaliciousURL. https://www.kaggle.com/antonyj453/ urldataset#data.csv. [Last accessed 11/2019].

[18] Kang Leng Chiew, Choon Lin Tan, KokSheik Wong, Kelvin SC Yong, and Wei King Tiong. 2019. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences 484 (2019), 153–166.

[19] Gerardo Canfora and Corrado Aaron Visaggio. 2016. A set of features to detect web security threats. Journal of Computer Virology and Hacking Techniques (2016).

[20] Eduardo Benavides, Walter Fuertes, Sandra Sanchez, and Manuel Sanchez. 2019. Classification of Phishing Attack Solutions by Employing Deep Learning Techniques: A Systematic Literature Review. In Developments and Advances in Defense and Security. Springer, 51–64.

[21] Sreyasee Das Bhattacharjee, Ashit Talukder, Ehab Al-Shaer, and Pratik Doshi. 2017. Prioritized active learning for malicious URL detection using weighted text-based features. In Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on. IEEE.

[22] Yazan Alshboul, Raj Nepali, and Yong Wang. 2015. Detecting malicious short URLs on Twitter. (2015).

[23] Betul Altay, Tansel Dokeroglu, and Ahmet Cosar. 2018. Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection. Soft Computing (2018).

[24] Ankesh Anand, Kshitij Gorde, Joel Ruben Antony Moniz, Noseong Park, Tanmoy Chakraborty, and Bei-Tseng Chu. 2018. Phishing URL detection with oversampling based on text generative adversarial networks. In 2018 IEEE International Conference on Big Data (Big Data). IEEE, 1168–1177.

[25] Farhan Douksieh Abdi and Lian Wenjuan. 2017. Malicious URL Detection using Convolutional Neural Network. Journal International Journal of Computer Science, Engineering and Information Technology (2017).

[26] R. Bharadwaj, A. Bhatia, L. D. Chhibbar, K. Tiwari and A. Agrawal, "Is this URL Safe: Detection of Malicious URLs Using Global Vector for Word Representation," 2022 International Conference on Information Networking (ICOIN), 2022, pp. 486-491, doi: 10.1109/ICOIN53446.2022.9687204.

[27] Jingbing Chen, Jie Yuan, Yuewei Li, Yiqi Zhang, Yufan Yang, and Ruiqi Feng. 2020. A Malicious Web Page Detection Model based on SVM Algorithm: Research on the Enhancement of SVM Efficiency by Multiple Machine Learning Algorithms. In 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2020). Association for Computing Machinery, New York, NY, USA, Article 51, 1–7. DOI:https://doi.org/10.1145/3446132.3446183

[28] Cláudio Marques, Silvestre Malta, João Paulo Magalhães,DNS dataset for malicious domains detection,Data in Brief,Volume 38,2021,107342,ISSN 2352-3409,https://doi.org/10.1016/j.dib.2021.107342.

[29] Jie Xu and Ao Ju. 2021. Detect Darknet URL Based on Artificial Neural Network. The 5th International Conference on Computer Science and Application Engineering. Association for Computing Machinery, New York, NY, USA, Article 57, 1–6. DOI:https://doi.org/10.1145/3487075.3487132

[30] Zhouyu Bao, Wenbo Wang, and Yuqing Lan. 2019. Using Passive DNS to Detect Malicious Domain Name. Proceedings of the 3rd International Conference on Vision, Image and Signal Processing. Association for Computing Machinery, New York, NY, USA, Article 85, 1–8. DOI:https://doi.org/10.1145/3387168.3387236

[31] A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment Brij B. Gupta, Krishna Yadav, Imran Razzak, Konstantinos Psannis, Arcangelo Castiglione, Xiaojun Chang.

[32] Detecting Malicious URLs in E-Mail – An Implementation. Dhanalakshmi Ranganayakulu, Chellappan.

[33] Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods Tie Li, Gang Kou, Yi Peng.

[34] Lexical features based malicious URL detection using machine learning techniques Saleem Raja A., Vinodini R., Kavitha A.

[35] Unsupervised malicious domain detection with less labeling effort Kyung Ho Park, Hyun Min Song, Jeong Do Yoo, Su-Youn Hong, Byoungmo Cho, Kwangsoo Kim, Huy Kang Kim.

[36] C. Ding, "Automatic Detection of Malicious URLs using Fine-Tuned Classification Model," 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), 2020, pp. 302-320, doi: 10.1109/ISCTT51595.2020.00060.

[37] A. Singh and P. K. Roy, "Malicious URL Detection using Multilayer CNN," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021, pp. 340-345, doi: 10.1109/3ICT53449.2021.9581880.

[38] S. -Y. Huang, T. -H. Chuang, S. -M. Huang and T. Ban, "Malicious URL Linkage Analysis and Common Pattern Discovery," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3172-3179, doi: 10.1109/BigData47090.2019.9006145.

[39] F. Sadique, R. Kaul, S. Badsha and S. Sengupta, "An Automated Framework for Real-time Phishing URL Detection," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0335-0341, doi: 10.1109/CCWC47524.2020.9031269.

[40] M. Arivukarasi and A. Antonidoss, "Performance Analysis of Malicious URL Detection by using RNN and LSTM," 2020 Fourth International Conference on Computing

Methodologies and Communication (ICCMC), 2020, pp. 454-458, doi: 10.1109/ICCMC48092.2020.ICCMC-00085.

[41] M. Akiyama, T. Yagi and T. Hariu, "Improved Blacklisting: Inspecting the Structural Neighborhood of Malicious URLs," in *IT Professional*, vol. 15, no. 4, pp. 50-56, July-Aug. 2013, doi: 10.1109/MITP.2012.118.