# MALEVOLENT AND PHISHING UNIFIED RESOURCE LOCATION DETECTION BASED ON MACHINE LEARNING DEEP LEARNING TECHNIQUES

[1]M.Parimala, [2]Dr.C.Srinivasa Kumar, [3]Dr.P.Rajendra Prasad

[1]Associate Professor, Department of CSE, Vignan's Institute of Management and Technology for Women, Kondapur, Ghatkesar, Telangana

[2]Professor & Dean, Department of CSE, Vignan's Institute of Management and Technology for Women, Kondapur, Ghatkesar, Telangana

[3]Associate Professor, Department of CSE, Vignan's Institute of Management and Technology for Women, Kondapur, Ghatkesar, Telangana

**Abstract**- Phishing attack is a simplest way to obtain sensitive information from innocent users. Aim of the phishers is to acquire critical information like username, password and bank account details. Currently, the risk of network information insecurity is increasing rapidly in number and level of danger. Domain phishing is a scam to trick email recipients into handing over their account details via links in emails posing as their registrar. A Phishing URL is a link created with the purpose of promoting scams, attacks, and frauds. When clicked on, Phishing URLs can download ransomware, lead to phishing or phishing emails, or cause other forms of cybercrime. The methods mostly used by hackers today is to attack end-to end technology and exploit human vulnerabilities. These techniques include social engineering, phishing, pharming, etc. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This project deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, Logistic Regression, GB and Support Vector Machine Algorithms are used to detect phishing websites. Aim of the project is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, false positive and false negative rate of each algorithm.

**Keywords:** *Machine Learning and Deep Learning Algorithms, Cyber Security, Malicious URL detection, Feature extraction, Feature selection.*

## 1. INTRODUCTION

The year 2020 saw peoples life being completely dependent on technology due to the global pandemic. Since digitalization became significant in this scenario, cyber criminals went on an internet crime spree. Recent reports and researches point to an increased number of security breaches that costs the victims a huge sum of money or disclosure of confidential data. Phishing is a cybercrime that employs both social engineering and technical subterfuge in order to steal personal identity data or financial account credentials of victims. In phishing, attackers counterfeit trusted websites and misdirect people to these websites, where they are tricked into sharing usernames, passwords, banking or credit card details and other sensitive credentials. These phishing URLs may be sent to the consumers through email, instant message or text message. According to the FBI crime report 2020, phishing was the most common type of cyber attack in 2020 and phishing incidents nearly doubled from 114,702 in 2019 to 241,342 in 2020. The Verizon 2020 Data Breach Investigation Report states that 22% of data breaches in 2020 involved phishing The number of phishing attacks as observed by the Anti- Phishing Work Group (APWG) grew through 2020, doubling.

### What is URL?

The Uniform Resource Locator (URL) is the well-defined structured format unique address for accessing websites over World Wide Web (WWW). Generally, there are three basic components that make up a legitimate URL

i.) Protocol: It is basically an identifier that determines what protocol to use e.g., HTTP, HTTPS, etc.

ii) Hostname: Also known as the resource name. It contains the IP address or the domain name where the actual resource is located.

iii) Path: It specifies the actual path where the resource is located

As per the figure, wisdomml.in.edu is the domain name. The top-level domain is another component of the domain name that tells the nature of the website i.e, commercial (.com), educational (.edu), organization (.edu), etc.

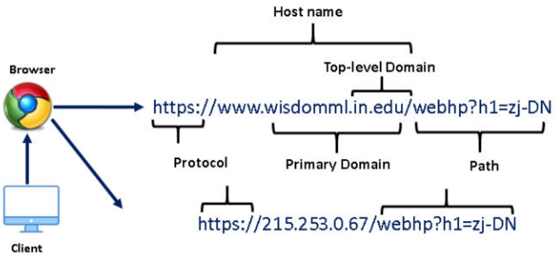

Host name
Top-level Domain
Browser
https://www.wisdomml.in.edu/webhp?h1=zj-DN
Protocol
Primary Domain
Path
https://215.253.0.67/webhp?h1=zj-DN
Client

Fig. 1. Components of a URL

**Figure 1:** Components Of URL

Malicious URL?

These type of URLs inject malware into the victim’s system once he/she visit such URLs. Modified or compromised

URLs employed for cyber attacks are known as malicious URLs.A malicious URL or website generally contains different types of trojans, malware, unsolicited content in the form of phishing, drive-by-download, spams. The main objective of the malicious website is to fraud or steal the personal or financial details of unsuspecting users. Due to the ongoing COVID-19 pandemic the incidents of cybercrime increased manifold. According to Symantec Internet Security Threat Report (ISTR) 2019, malicious URLs are a highly used technique in cyber crimes. Phishing Detection

A URL based phishing attack is carried out by sending malicious links, that seems legitimate to the users, and tricking them into clicking on it. In phishing detection, an incoming URL is identified as phishing or not by analysing the different features of the URL and is classified accordingly. Different machine learning algorithms are trained on various datasets of URL features to classify a given URL as phishing or legitimate.
Phishing Detection Approaches

In List Based approach, there are two lists, called whitelist and blacklist to classify legitimate and phishing URLs respectively. In access to websites takes place only if the URL is in the whitelist. In blacklist is used. In Heuristic Based approach, the structure of a phishing URL is analysed. A pattern of URLs that were previously classified as phishing is created. URLs are classified according to their compliance with this pattern. The methods used to process the features of the URL plays a significant role in classifying websites accurately. Visual similarity Based approach works by comparing the visual similarity of the website pages. Websites are classified as phishing or not by taking a server side view of them as in. These two data are then compared with image processing techniques. Fake web pages are designed very close to the original ones and it is easier to notice minor differences with image processing techniques, as users cannot notice them easily. Content Based approach analyses the pages content. This method extracts features from page contents and third-party services like search engines and DNS servers. In authors proposed a detection method by specifying weights to the words that draw out from URLs and HTML contents. The words might include brand names that attackers use in the URL to make it look like a real one. Weights are specified according to their presence at different positions in URLs. The most probable words are chosen and then sent to Yahoo search to return the domain name with the highest frequency between the top 30 outcomes. The owners of the domain name are compared to decide if the website is phishing or not. In they utilized a logo image to find the identity of web pages by matching real and fake web pages.

.

## II Literature Survey

In this section, few of the research works that deploy the above mentioned algorithms are reviewed and their results are summarized.

In the paper [12], the authors Rishikesh Mahajan and Irfan Siddavatam chose three algorithms for classification Decision Tree, Random Forest and Support Vector Machine. Their dataset contained 17,058 benign URLs and 19,653 phishing URLs collected from Alexa website and PhishTank respectively, with 16 features each. The dataset was divided into training and testing set in the ratios 50:50, 70:30 and 90:10 respectively. The accuracy score, false negative rate and false positive rate were considered as performance evaluation metrics. They achieved 97.14% accuracy for Random Forest algorithm with the lowest false negative rate. The paper concluded that accuracy increases when more data is used for training.

The study conducted by Jitendra Kumar et al. in [13] trained different classifiers like Logistic Regression, Naive Bayes Classifier, Random Forest, Decision Tree and K- Nearest Neighbor based on the features extracted from the lexical structure of the URL. They created the dataset of URLs in such a way that it solved the issues of data imbalance, biased training, variance and overfitting. The dataset contained an equal number of labeled phishing and legitimate URLs, and was further split in the ratio 7:3 for training and testing. All the classifiers had almost the same AUC (area under ROC curve), but the Naive Bayes Classifier turned out to be more suitable as it had the highest AUC value. Naive Bayes achieved the highest accuracy of 98% with a precision=1, recall=0.95 and F1-score=0.97.

Mehmet Korkmaz et al. proposed in [14] a machine- learning based phishing detection system by using 8 different algorithms on three different datasets. The algorithms used were Logistic Regression (LR), K-Nearest Neighbor(KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), XGBoost, Random Forest (RF) and Artificial Neural Network (ANN). It was observed that the models using LR, SVM and NB have low accuracy rate. In terms of training time, NB, DT, LR and ANN algorithms gave better results. They concluded that RF algorithm or ANN algorithm may be used because of less training time along with a high accuracy rate.

Mohammad Nazmul Alam et al. [15] proposed a system to detect phishing attacks using Random Forest and Decision Tree. The Kaggle dataset with 32 features was used along with feature selection algorithms like principal component analysis (PCA). Feature selection reduces redundancy of data that is irrelevant or unnecessary in the dataset. The proposed model used REF, Relief-F, IG and GR algorithm for feature selection before applying PCA. Random Forest achieved an accuracy of 97%. It had less variance, and it could handle the over-fitting problem.

Abdulhamit Subasi et al. in [16] presented an intelligent phishing detection system using UCI dataset. Different machine learning tools namely, Artificial Neural Networks (ANN), K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), C4.5 Decision Tree, Random Forest (RF), and Rotation Forest (RoF) were used as classifiers for detection of phishing websites. The performance of proposed RF classifier was higher than the others in terms of accuracy, F-measure and AUC. RF was faster, robust and more accurate than the other classifiers.
Today, android smartphones are being used by billions of users and thus have become a lucrative target of malware designers. Therefore being one step ahead in this zero-sum game of malware detection between the anti-malware community and malware developers is more of a necessity than a desire. This work focuses on a proactive adversary-aware framework to develop adversarially superior android malware detection

models. We first investigate the adversarial robustness of thirty-six distinct malware detection models constructed using two static features (permission and intent) and eighteen classification algorithms. We designed two Targeted Type-II Evasion Attacks (TRPO-MalEAttack and PPO-MalEAttack) based on reinforcement learning to exploit vulnerabilities in the above malware detection models. The attacks aim to add minimum perturbations in each malware application and convert it into an adversarial application that can fool the malware detection models. The TRPO-MalEAttack achieves an average fooling rate of 95.75% (with 2.02 mean perturbations), reducing the average accuracy from 86.01% to 49.11% in thirty-six malware detection models. On the other hand, The PPO-MalEAttack achieves a higher average fooling rate of 96.87% (with 2.08 mean perturbations), reducing the average accuracy from 86.01% to 48.65% in the same thirty-six detection models. We also develop a list of the TEN most vulnerable android permissions and intents that an adversary can use to generate more adversarial applications. Later, we propose a defense strategy (MalVPatch) to counter the adversarial attacks on malware detection models. The MalVPatch defense achieves higher detection accuracy along with a drastic improvement in the adversarial robustness of malware detection models. Finally, we conclude that investigating the adversarial robustness of models is necessary before their real-world deployment and helps achieve adversarial superiority in android malware detection

Recent years have witnessed a significant increase in the use of Android devices in many aspects of our life. However, users can download Android apps from third-party channels, which provides numerous opportunities for malware. Attackers utilize unsolicited permissions to gain access to the sensitive private intelligence of users. Since signature-based antivirus solutions no longer meet practical needs, efficient and adaptable solutions are desperately needed, especially in new variants. As a remedy, we propose a hybrid Android malware detection approach that combines dynamic and static tactics. We firstly adopt static analysis inferring different permission usage patterns between malware and benign apps based on the machine-learning-based method. To classify the suspicious apps further, we extract the object reference relationships from the memory heap to construct a dynamic feature base. We then present an improved state-based algorithm based on DAMBA. Experimental results on a real-world dataset of 21,708 apps show that our approach outperforms the well-known detector with 97.5% F1-measure. Besides, our system is demonstrated to resist permission abuse behaviors and obfuscation techniques.

Since the development of information systems during the last decade, cybersecurity has become a critical concern for many groups, organizations, and institutions. Malware applications are among the commonly used tools and tactics for perpetrating a cyberattack on Android devices, and it is becoming a challenging task to develop novel ways of identifying them. There are various malware detection models available to strengthen the Android operating system against such attacks. These malware detectors categorize the target applications based on the patterns that exist in the features present in the Android applications. As the analytics data continue to grow, they negatively affect the Android defense mechanisms. Since large numbers of unwanted features create a performance bottleneck for the detection mechanism, feature selection techniques are found to be beneficial. This work presents a Rock Hyrax Swarm Optimization with deep learning-based Android malware detection (RHSODL-AMD) model. The technique presented includes finding the Application Programming Interfaces (API) calls and the most significant permissions, which results in effective discrimination between the good ware and malware applications. Therefore, an RHSO based feature subset selection (RHSO-FS) technique is derived to improve the classification results. In addition, the Adamax optimizer with attention recurrent autoencoder (ARAE) model is employed for Android malware detection. The experimental validation of the RHSODL-AMD technique on the Andro-AutoPsy dataset exhibits its promising performance, with a maximum accuracy of 99.05%

## III. PROPOSED SYSTEM

The increased usage of cloud services, growing number of users, changes in network infrastructure that connect devices running mobile operating systems, and constantly evolving network technology cause novel challenges for cyber security that have never been foreseen before. Cyber security is a very important requirement for users. With the rise in Internet usage in recent years, cyber security has become a serious concern for computer systems. When a user accesses a malicious and phishing Web site, it initiates a malicious behavior that has been pre-programmed. As a result, there are numerous methods for locating potentially hazardous URLs on the Internet. Traditionally, detection was based heavily on the usage of blacklists. Blacklists, on the other hand, are not exhaustive and cannot detect newly created harmful URLs. Recently, machine learning methods have received a lot of importance as a way to improve the majority of malicious URL detectors. The main goal of this research is to compile a list of significant features that can be utilized to detect and classify the majority of malicious URLs. To increase the effectiveness of classifiers for detecting malicious URLs, this study recommends utilizing host-based and lexical aspects of the URLs.

Phishing is a social engineering cyberattack where criminals deceive users to obtain their credentials through a login form that submits the data to a malicious server. In this paper, we compare machine learning and deep learning techniques to present a method capable of detecting phishing websites through URL analysis. In most current state-of-the-art solutions dealing with phishing detection, the legitimate class is made up of homepages without including login forms. On the contrary, we use URLs from the login page in both classes because we consider it is much more representative of a real case scenario and we demonstrate that existing techniques obtain a high false-positive rate when tested with URLs from legitimate login pages. Additionally, we use datasets from different years to show how models decrease their accuracy over time by training a base model with old datasets and testing it with recent URLs. Also, we perform a frequency analysis over current phishing domains to identify different techniques carried out by phishers in their campaigns. To prove these statements, we have created a new dataset named Phishing Index Login URL (PILU-90K), which is composed of 60K legitimate URLs, including index and login websites, and 30K phishing URLs. Finally, we present a Logistic Regression model which, combined with Term Frequency - Inverse Document Frequency (TF-IDF) feature extraction, obtains 96:50% accuracy on the introduced login URL dataset.

A popular approach in detecting malicious activity on the web is by leveraging distinguishing features between malicious and benign DNS usage. Both passive DNS monitoring and active DNS probing methods have been used to identify malicious domains. While some of these efforts focused solely on detecting fast flux service networks, another can also detect domains implementing phishing and drive-by-downloads. The best-known non-proprietary content-based approach to detect phishing webpages is Cantina

**Disadvantages**

•Existing tools such as Google Safe Browsing are not enabled on the mobile versions of browsers, thereby precluding mobile users.
•DNS based mechanisms do not provide deeper understanding of the specific activity implemented by a webpage or domain.
•Downloading and executing each webpage impacts performance and hinders scalability of dynamic approaches.
•URL-based techniques usually suffer from high false positive rates.
•Cantina suffers from performance problems due to the time lag involved in querying the Google search engine. Moreover, Cantina does not work well on webpages written in languages other than English.
•Finally, existing techniques do not account for new mobile threats such as known fraud phone numbers that attempt to trigger the dialer on the phone

In the proposed system, machine learning algorithms are used to classify URLs based on the features and behaviors of URLs. The features are extracted from static and dynamic behaviors of URLs and are new to the literature. Those newly proposed features are the main contribution of the research. Machine learning algorithms are a part of the whole malicious and phishing URL detection system. Two supervised machine learning algorithms are used, Support vector machine (SVM) and Random forest (RF).

Advantages

•Protection from malicious attacks on your network.
•Deletion and/or guaranteeing malicious elements within a preexisting network.
•Prevents users from unauthorized access to the network.
•Deny's programs from certain resources that could be infected.
•Securing confidential information
•The proposed algorithms are suitable to utilized the usefulness of our new features selected for malicious URL detection.
•In the proposed work, SVM and RF are selected as an example to illustrate the good performance of the whole detection system, and are not our main focus. Readers are encouraged to implement some other algorithms such as Naïve Bayes, Decision trees, k-nearest neighbors, neural networks, etc.

## Machine Learning Classifiers

### RandomForestClassifier

RandomForestClassifier can analyze the importance of different features in distinguishing between benign and malicious applications. By examining feature importance scores, analysts can gain insights into the characteristics and behaviors that are most indicative of malware presence. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

A random forest algorithm is used to classify the features after they have been extracted. If we break down the word, it consists of forest, which is a collection of decision trees, and random, which refers to the fact that we are sampling at random. When this approach is applied to a data set, a portion of the data is used as a training set, and the data is clustered into groups and subgroups. A decision tree is a structure that looks like a tree and is created by connecting data points to groups and sub-groups. The program then creates a forest out of several trees. However, each tree is unique since the variables are chosen at random for each split in the tree. Apart from the training set, the remaining data is utilized to forecast which tree in the forest produces the best categorization of data points, and the tree with the highest predictive power is displayed as output. The type of each program is then determined using a set of labels, with 1 denoting malware and 0 denoting benign files. By minimizing the uncertainty of the class labels, the decision tree splits the training set into two subsets with distinct labels at each node.

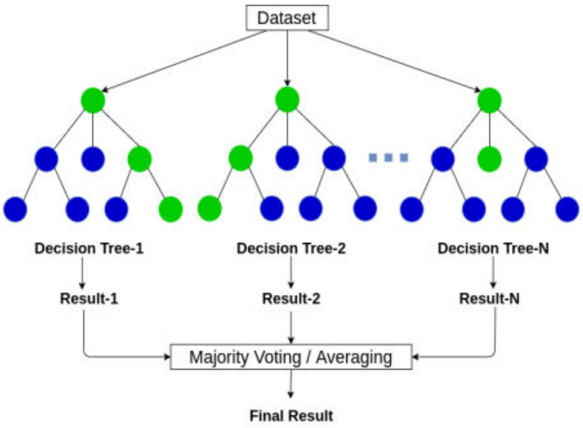


Fig 2: Random Forest Classifier

Linear regression is a supervised machine learning algorithm used to determine the linear relationship between a dependent variable and one or more independent features. When there is only one independent feature, it is termed Univariate Linear Regression. If there are multiple features, it is called Multivariate Linear Regression. The primary goal of linear regression is to find the best fit line, minimizing the error between predicted and actual values. The best fit line equation defines a straight line representing the relationship between dependent and independent variables. The slope of this line signifies the extent to which the dependent variable changes for a unit change in the independent variable(s). The objective is to minimize errors along this best fit line.

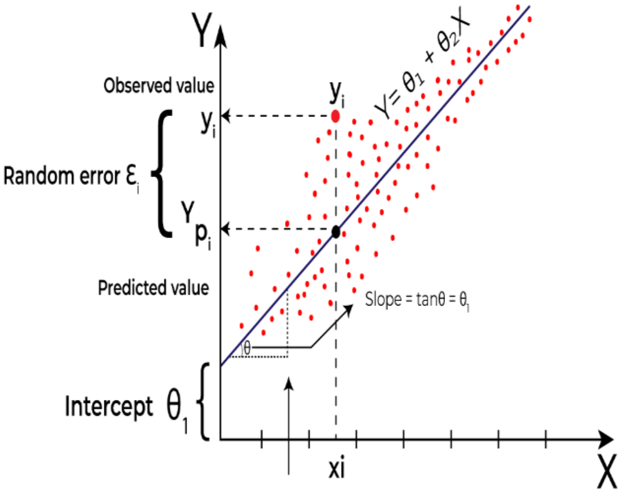

Y
Observed value
$y_i$
Random error $\varepsilon_i$
$Y_{p_i}$
Predicted value
$y_i$
$Y = \theta_1 + \theta_2 X$
Slope = tanθ = $\theta_1$
θ
Intercept $\theta_1$
xi
X

Fig 3: SVN Classifier

In the context of linear regression, Y is commonly referred to as the dependent or target variable, while X is known as the independent variable, also recognized as the predictor of Y. Linear regression encompasses various functions or modules for regression tasks, with the linear function being the simplest among them. The independent variable X can represent either a single feature or multiple features relevant to the problem at hand. Linear regression is designed to predict the value of the dependent variable (Y) based on a given independent variable (X). Consequently, the term "Linear Regression" derives from this predictive relationship. For example, in a scenario where X represents work experience and Y represents salary, the regression line serves as the best fit line for our model. To determine the best fit line, we rely on a cost function. This function assists in computing the optimal values necessary to obtain the best fit line. Given that different weights or coefficients of lines lead to distinct regression lines, the cost function aids in identifying the most suitable parameters for the model.

## IV METHODOLOGY

The implemented system has demonstrated efficacy in both detecting and preventing botnet attacks. Through rigorous testing and data collection, it has proven capable of passively monitoring sensor data and issuing alerts in real-time upon detection of an attack. Leveraging this feedback, the system feeds the data into an attribution model to ascertain the attack's attributes. Subsequently, security experts and incident response teams utilize the framework's efficient and accurate information to promptly address detected attacks and proactively prevent potential damages.

System Model

In the first module, we develop the System environment model. Website providers use JavaScript or user agent strings to identify and then redirect mobile users to a mobile specific version. We note that not all static features used in existing techniques differ when measured on mobile and desktop webpages.Mobile websites enable access to a user's personal information and advanced capabilities of mobile devices through web APIs. Existing static analysis techniques do not consider these mobile specific functionalities in their feature set.We argue and later demonstrate that accounting for the mobile specific functionalities helps identify new threats specific to the mobile web. For example, the presence of a known 'bank' fraud number on a website might indicate that the webpage is a phishing webpage imitating the same bank

Malicious Pages

We argue that benign webpage writers take effort to provide good user experience, whereas the goal for malicious webpage authors is to trick users into performing intentional actions with minimal effort. We therefore examine whether a webpage has no script content admeasure the number of no script. Intuitively, a benign webpage writer will have more noscript in the code toensure good experience even for a security savvy user.

Identifying relevant static features

We extract static features from a webpage and make predictions about its potential maliciousness. We first discuss the feature set used in fraud detection followed by the collection process of the dataset. Structural and lexical properties of a URL have been used to differentiate between malicious and benign webpages. However, using only URL features for such differentiation leads to a high false positive rate.

Our data gathering process included accumulating labeled benign and malicious mobile specific webpages .First, we describe an experiment that identifies and defines 'mobile specific webpages. We then conduct the data collection process. We use these crawls specifically because they are close to the publication of the related work, making them as close to equivalent as possible.

Detect malicious webpages

We describe the machine learning techniques we considered to tackle the problem of classifying mobile specific webpages as malicious or benign. We then discuss the strengths and weaknesses of each classification technique, and the process for selecting the best model for fraud detection. We build and evaluate our chosen model for accuracy, false positive rate and true positive rate. Finally, we compare fraud detection to existing techniques and empirically demonstrate the significance of fraud detection's features. We note that where automated analysis is possible, we use our full datasets; however, as is commonly done in the research community, we use randomly selected subsets of our data when extensive manual analysis and verification is required.

The observations obtained from the survey are pointed out in Section VII. Section VIII concludes the paper.

## DATASETS

Usually, the phishing website data is collected from Phish Tank or OpenPhish. PhishTank.com is a website where phishing URLs are detected and can be accessed via API call. Their data is used by companies like McAfee, Kaspersky, Mozilla and APWG. Since it does not store the content of webpages, it is a good source for URL based analysis. The legitimate sites are generally collected from Alexas top- ranking websites database or from common-crawl. There are publicly available datasets like the UCI machine learning repository dataset used in which contains 11,055 records, each record having 31 features and the Kaggle phishing dataset used in .

## FEATURE EXTRACTION

URLs have certain characteristics and patterns that can be considered as its features. The Fig. 3 shows the relevant parts of a typical URL. In case of URL based analysis for designing machine learning models, we need to extract these features in order to form a dataset that can be used for training and testing. There are four categories of features that are most commonly considered for feature extraction as in. They are as follows:

1. Address Bar based features
2. Abnormal based features
3. HTML and JavaScript based features
4. Domain based features

## PHISHING URL DETECTION RULES

•The system designs the following concepts which Presence of IP address in URL: If IP address present in URL then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that attacker is trying to steal sensitive information.

•Presence of @ symbol in URL: If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add special symbol @ in the URL leads the browser to ignore everything preceding the "@" symbol and the realaddress often follows the "@" symbol].

•Number of dots in Hostname: Phishing URLs have many dots in URL. For example http://shop.fun.amazon.phishing.com, in this URL phishing.com is an actual domain name, whereas use of "amazon" word is to trick users to click on it. Average number of dots in benign URLs is 3. If the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.

•Prefix or Suffix separated by (-) to domain: If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example Actual site is http://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.com to confuse the innocent users.

•URL redirection: If "//" present in URL path then feature is set to 1 else to 0. The existence of "//" within the URL path means that the user will be redirected to another website [4].

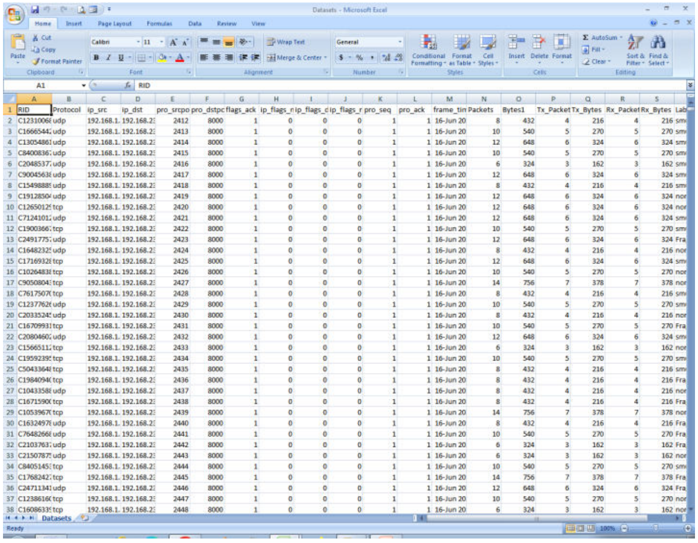
•HTTPS token in URL: If HTTPS token present in URL then the feature is set to 1 else to 0. Phishers may add the "HTTPS" token to the domain part of a URL in order to trick users. For example, http://https-wwwpaypal-it-mpp-home.soft-hair.com [4].

•Information submission to Email: Phisher might use "mail()" or "mailto:" functions to redirect the user's information to his personal email[4]. If such functions are present in the URL then feature is set to 1 else to 0.

•URL Shortening Services "TinyURL": TinyURL service allows phisher to hide long phishing URL by making it short. The goal is to redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0

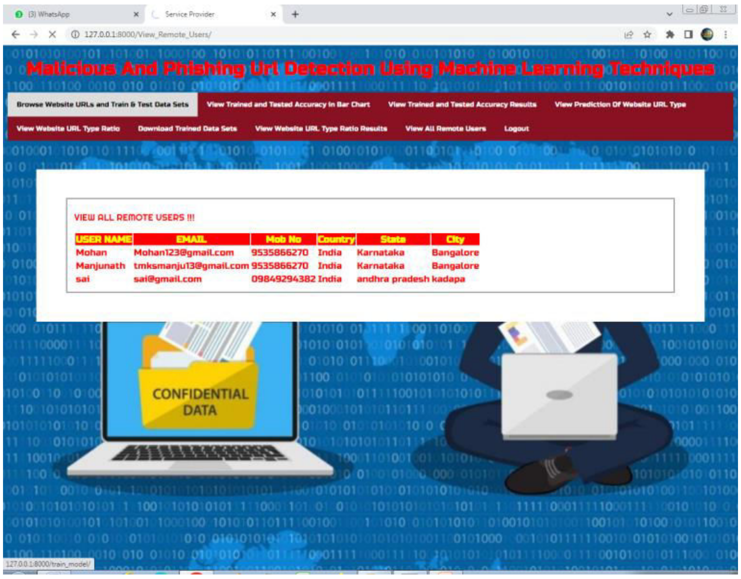
•Length of Host name: Average length of the benign URLs is found to be a 25, If URL's length is greater than 25 then the feature is set to 1 else to 0.

•Presence of sensitive words in URL: Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that found in many phishing URLs :- 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal', 'password', 'username', etc;



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | RID | Protocol | ip_src | ip_dst | pro_srcpo | pro_dstpc | flags_ack | ip_flags_n | ip_flags_d | ip_flags_r | pro_seq | pro_ack | frame_tin | Packets | Bytes1 | Tx_Packet | Tx_Bytes | Rx_Packet | Rx_Bytes | Lab |
| 2 | C12310068 | udp | 192.168.1. | 192.168.2 | 2412 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | sm |
| 3 | C16665442 | udp | 192.168.1. | 192.168.2 | 2413 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 4 | C13054861 | udp | 192.168.1. | 192.168.2 | 2414 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | sm |
| 5 | C84008367 | udp | 192.168.1. | 192.168.2 | 2415 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 6 | C20485377 | udp | 192.168.1. | 192.168.2 | 2416 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 6 | 324 | 3 | 162 | 3 | 162 | sm |
| 7 | C90045638 | udp | 192.168.1. | 192.168.2 | 2417 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | sm |
| 8 | C15498889 | udp | 192.168.1. | 192.168.2 | 2418 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | sm |
| 9 | C19128504 | udp | 192.168.1. | 192.168.2 | 2419 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | nor |
| 10 | C12650125 | tcp | 192.168.1. | 192.168.2 | 2420 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | nor |
| 11 | C71241012 | udp | 192.168.1. | 192.168.2 | 2421 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | sm |
| 12 | C19003667 | tcp | 192.168.1. | 192.168.2 | 2422 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 13 | C24917757 | udp | 192.168.1. | 192.168.2 | 2423 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | Fra |
| 14 | C16482325 | udp | 192.168.1. | 192.168.2 | 2424 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | nor |
| 15 | C17169328 | tcp | 192.168.1. | 192.168.2 | 2425 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | sm |
| 16 | C10264838 | tcp | 192.168.1. | 192.168.2 | 2426 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | nor |
| 17 | C90508043 | tcp | 192.168.1. | 192.168.2 | 2427 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 14 | 756 | 7 | 378 | 7 | 378 | nor |
| 18 | C76175070 | tcp | 192.168.1. | 192.168.2 | 2428 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | sm |
| 19 | C12377626 | udp | 192.168.1. | 192.168.2 | 2429 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 20 | C20335245 | udp | 192.168.1. | 192.168.2 | 2430 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | nor |
| 21 | C16709931 | tcp | 192.168.1. | 192.168.2 | 2431 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | Fra |
| 22 | C20804602 | udp | 192.168.1. | 192.168.2 | 2432 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | sm |
| 23 | C15665112 | tcp | 192.168.1. | 192.168.2 | 2433 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 6 | 324 | 3 | 162 | 3 | 162 | nor |
| 24 | C19592395 | tcp | 192.168.1. | 192.168.2 | 2434 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 25 | C50433648 | tcp | 192.168.1. | 192.168.2 | 2435 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | sm |
| 26 | C19840940 | tcp | 192.168.1. | 192.168.2 | 2436 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | Fra |
| 27 | C10433588 | udp | 192.168.1. | 192.168.2 | 2437 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | nor |
| 28 | C16715900 | tcp | 192.168.1. | 192.168.2 | 2438 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | Fra |
| 29 | C10539670 | tcp | 192.168.1. | 192.168.2 | 2439 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 14 | 756 | 7 | 378 | 7 | 378 | nor |
| 30 | C16324978 | udp | 192.168.1. | 192.168.2 | 2440 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 8 | 432 | 4 | 216 | 4 | 216 | Fra |
| 31 | C76482668 | udp | 192.168.1. | 192.168.2 | 2441 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | Fra |
| 32 | C21037637 | udp | 192.168.1. | 192.168.2 | 2442 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 6 | 324 | 3 | 162 | 3 | 162 | Fra |
| 33 | C21507875 | udp | 192.168.1. | 192.168.2 | 2443 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 6 | 324 | 3 | 162 | 3 | 162 | nor |
| 34 | C84051453 | tcp | 192.168.1. | 192.168.2 | 2444 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | sm |
| 35 | C17682427 | tcp | 192.168.1. | 192.168.2 | 2445 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 14 | 756 | 7 | 378 | 7 | 378 | Fra |
| 36 | C24711341 | udp | 192.168.1. | 192.168.2 | 2446 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 12 | 648 | 6 | 324 | 6 | 324 | Fra |
| 37 | C12386160 | tcp | 192.168.1. | 192.168.2 | 2447 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 10 | 540 | 5 | 270 | 5 | 270 | nor |
| 38 | C16086335 | tcp | 192.168.1. | 192.168.2 | 2448 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 20 | 6 | 324 | 3 | 162 | 3 | 162 | nor |

**Fig 4: Router with Various Routes**

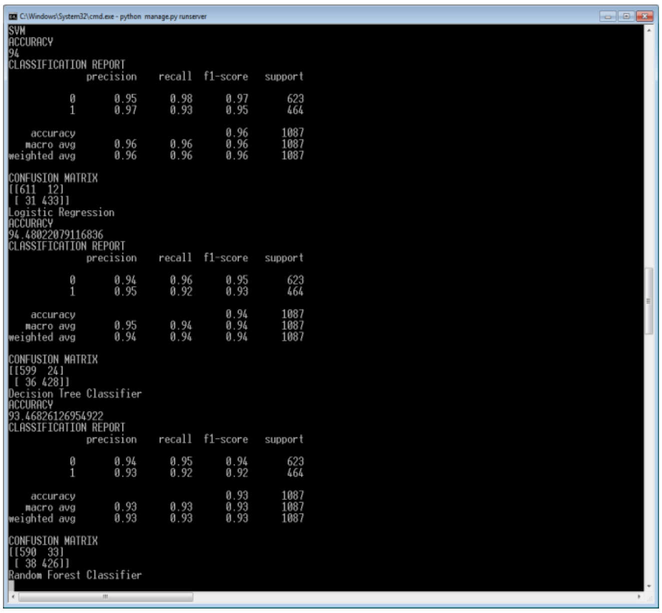

(3) WhatsApp
Service Provider
127.0.0.1:8000/View_Remote_Users/
Browse Website URLs and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Website URL Type
View Website URL Type Ratio
Download Trained Data Sets
View Website URL Type Ratio Results
View All Remote Users
Logout
CONFIDENTIAL DATA
127.0.0.1:8000/train_model/

Malicious And Phishing Url Detection Using Machine Learning Techniques

VIEW ALL REMOTE USERS !!!

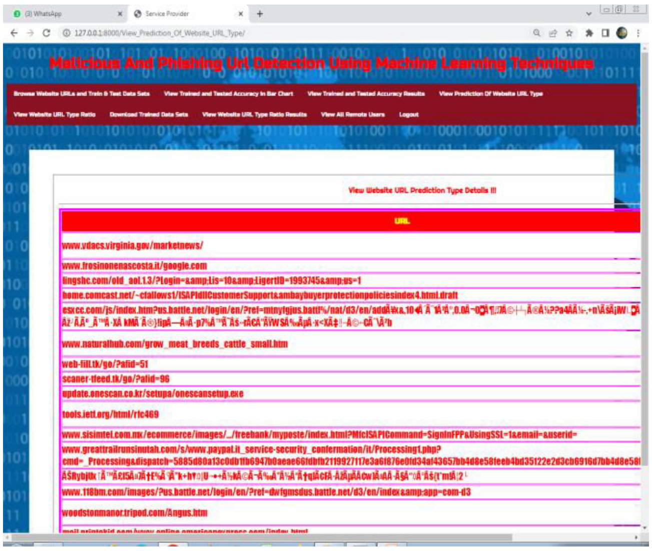
| USER NAME | EMAIL | Mob No | Country | State | City |
|---|---|---|---|---|---|
| Mohan | Mohan123@gmail.com | 9535866270 | India | Karnataka | Bangalore |
| Manjunath | tmksmanju13@gmail.com | 9535866270 | India | Karnataka | Bangalore |
| sai | sai@gmail.com | 09849294382 | India | andhra pradesh | kadapa |

**Fig 5: ML Classifiers Showing Acuracy**



```
SVM
ACCURACY
94
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       0.95      0.98      0.97       623
           1       0.97      0.93      0.95       464

    accuracy                           0.96      1087
   macro avg       0.96      0.96      0.96      1087
weighted avg       0.96      0.96      0.96      1087

CONFUSION MATRIX
[[611  12]
 [ 31 433]]
Logistic Regression
ACCURACY
94.48022079116836
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       623
           1       0.95      0.92      0.93       464

    accuracy                           0.94      1087
   macro avg       0.95      0.94      0.94      1087
weighted avg       0.94      0.94      0.94      1087

CONFUSION MATRIX
[[599  24]
 [ 36 428]]
Decision Tree Classifier
ACCURACY
93.46826126954922
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       0.94      0.95      0.94       623
           1       0.93      0.92      0.92       464

    accuracy                           0.93      1087
   macro avg       0.93      0.93      0.93      1087
weighted avg       0.93      0.93      0.93      1087

CONFUSION MATRIX
[[590  33]
 [ 38 426]]
Random Forest Classifier
```

Fig 6: ML Accuracy with Precision , Recall Score


(3) WhatsApp
Service Provider
127.0.0.1:8000/View_Prediction_Of_Website_URL_Type/
Malicious And Phishing Url Detection Using Machine Learning Techniques
Browse Website URLs and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Website URL Type
View Website URL Type Ratio
Download Trained Data Sets
View Website URL Type Ratio Results
View All Remote Users
Logout

View Website URL Prediction Type Details !!!

| URL |
|---|
| www.vdacs.virginia.gov/marketnews/ |
| www.frosinonenascosta.it/google.com |
| lingshc.com/old_aol.1.3/?Login=&amp;Lis=10&amp;LigertID=1993745&amp;us=1 |
| home.comcast.net/~cfallows1/ISAPIdllCustomerSupport&ambaybuyerprotectionpoliciesindex4.html.draft |
| esxcc.com/js/index.htm?us.battle.net/login/en/?ref=mtnyfgjus.batt%/nat/d3/en/add[illegible] |
| www.naturalhub.com/grow_meat_breeds_cattle_small.htm |
| web-fill.tk/go/?afid=51 |
| scaner-tfeed.tk/go/?afid=96 |
| update.onescan.co.kr/setupa/onescansetup.exe |
| tools.ietf.org/html/rfc469 |
| www.sisimtel.com.mx/ecommerce/images/../freebank/myposte/index.html?MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid= |
| www.greattrailrunsinutah.com/s/www.paypal.it_service-security_confermation/it/Processing1.php?cmd=_Processing&dispatch=5885d80a13c0db1fb6947b0aeae66fdbfb2119927117e3a6f876e0fd34af43657bb4d8e58feeb4bd35122e2d3cb6916d7bb4d8e58 |
| [illegible] |
| www.118bm.com/images/?us.battle.net/login/en/?ref=dwfgmsdus.battle.net/d3/en/index&amp;app=com-d3 |
| woodstonmanor.tripod.com/Angus.htm |

Fig 9: Tabular View of Botnet Dos Attack Detection for uploaded Data.

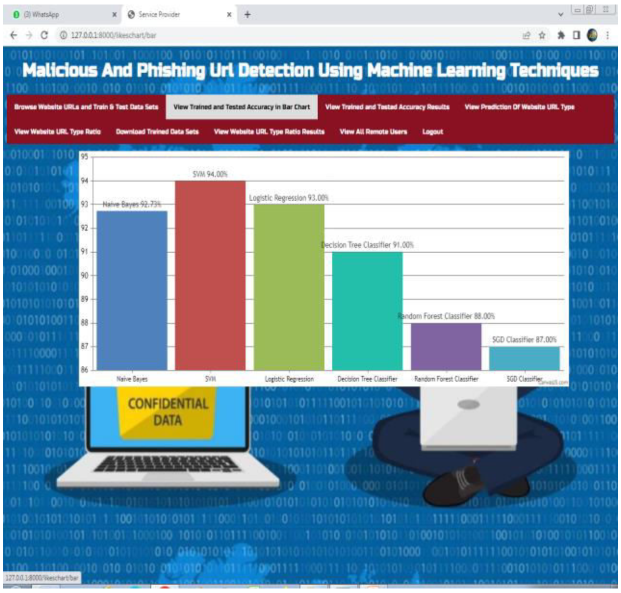
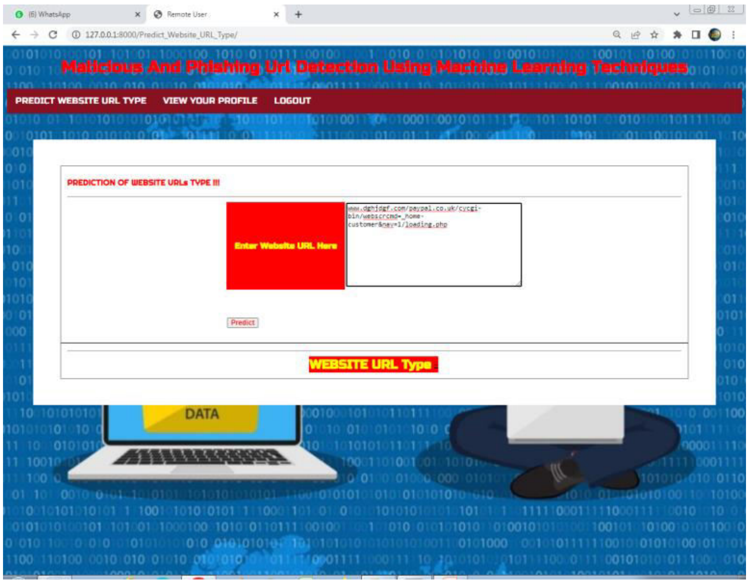


Fig 7: Bar Graph Showing Accuracy of Project

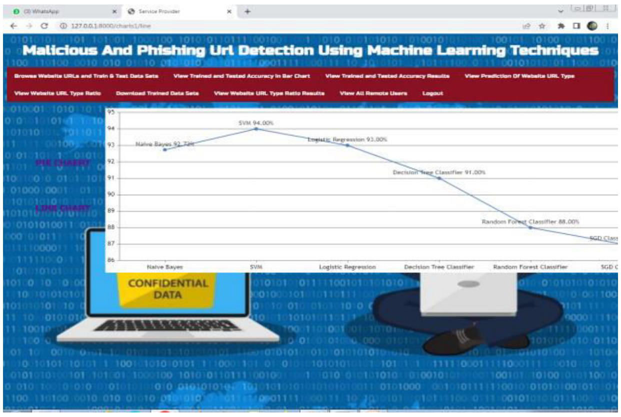

(6) WhatsApp
Remote User
127.0.0.1:8000/Predict_Website_URL_Type/
Malicious And Phishing Url Detection Using Machine Learning Techniques
PREDICT WEBSITE URL TYPE
VIEW YOUR PROFILE
LOGOUT
Predict
DATA

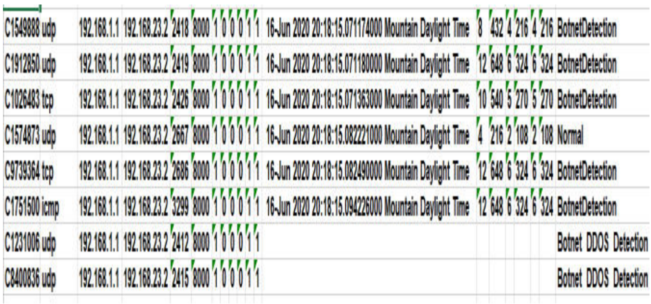
PREDICTION OF WEBSITE URLs TYPE !!!

Enter Website URL Here: www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrcmd=_home-customer&nav=1/loading.php

WEBSITE URL Type

Fig 9: Prediction of Attacks





| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C1549888 | udp | 192.168.1.1 | 192.168.23.2 | 2418 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.071174000 Mountain Daylight Time | 8 | 432 | 4 | 216 | 4 | 216 | BotnetDetection |
| C1912850 | udp | 192.168.1.1 | 192.168.23.2 | 2419 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.071180000 Mountain Daylight Time | 12 | 648 | 6 | 324 | 6 | 324 | BotnetDetection |
| C1026483 | tcp | 192.168.1.1 | 192.168.23.2 | 2426 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.071363000 Mountain Daylight Time | 10 | 540 | 5 | 270 | 5 | 270 | BotnetDetection |
| C1574873 | udp | 192.168.1.1 | 192.168.23.2 | 2667 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.082221000 Mountain Daylight Time | 4 | 216 | 2 | 108 | 2 | 108 | Normal |
| C9739364 | tcp | 192.168.1.1 | 192.168.23.2 | 2686 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.082490000 Mountain Daylight Time | 12 | 648 | 6 | 324 | 6 | 324 | BotnetDetection |
| C1751500 | icmp | 192.168.1.1 | 192.168.23.2 | 3299 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | 16-Jun 2020 20:18:15.094226000 Mountain Daylight Time | 12 | 648 | 6 | 324 | 6 | 324 | BotnetDetection |
| C1231006 | udp | 192.168.1.1 | 192.168.23.2 | 2412 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | | | | | | | | Botnet DDOS Detection |
| C8400836 | udp | 192.168.1.1 | 192.168.23.2 | 2415 | 8000 | 1 | 0 | 0 | 0 | 1 | 1 | | | | | | | | Botnet DDOS Detection |

Fig 10: Prediction Attack Results

## V. CONCLUSION

Phishing detection is now an area of great interest among the researchers due to its significance in protecting privacy and providing security. There are many methods that perform phishing detection by classification of websites using trained machine learning models. URL based analysis increases the speed of detection. Furthermore, by applying feature selection algorithms and dimensionality reduction techniques, we can reduce the number of features and remove irrelevant data. There are many machine learning algorithms that perform classification with good performance measures. In this paper, we have done a study of the process of phishing detection and the phishing detection schemes in the recent research literature. This will serve as a guide for new researchers to understand the process and to develop more accurate phishing detection systems

Further present a multiparty access control mechanism over the cipher text, which allows the data co-owners to append their access policies to the cipher text. Besides, we provide three policy aggregation strategies including full permit, owner priority and majority permit to solve the problem of privacy conflicts. In the future, we will enhance our scheme by supporting keyword search over the cipher text

## REFERENCE

1. Anti-phishing Working Group (APWG) Phishing Activity Trends Report 4th quarter 2020, https://docs.apwg.org/reports/apwg trends report q4 2020.pdf

2. FBI Internet Crime Report 2020, https://www.ic3.gov/Media/PDF/AnnualReport/2020 IC3Report.pdf

3. Verizon 2020 Data Breach Investigation Report, https://enterprise.verizon.com/resources/reports/2020-data-breachinvestigations- report.pdf

4. World Health Organization, Communicating for Health, Cyber Security, https://www.who.int/about/communications/cyber-security

5. Ye Cao, Weili Han, and Yueran Le, Anti-phishing based on automated individual white-list, Proceedings of the 4th ACM workshop on Digital identity management-DIM 08, pp. 51-60, 2008

6. M. Sharifi, and S. H. Siadati, A phishing sites blacklist generator, 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 840-843, 2008

7. N. Abdelhamid, A. Ayesh, and F. Thabtah, Phishing detection based associative classification data mining, Expert Systems with Applications, vol. 41, no.13, pp. 5948-5959, 2014

8. L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, Detection of phishing webpages based on visual similarity, Special interest tracks and posters of the 14th international conference on World Wide Web-WWW 05, pp. 1060-1061, 2005

9. C. L. Tan, K. L. Chiew et al., Phishing website detection using url assisted brand name weighting system, 2014 International Symposium on Intelligent Signal Processing and Communication Systems(ISPACS), IEEE, pp. 054-059, 2014

[10] Anti-phishing Working Group (APWG) Phishing Activity Trends Report 4th quarter 2020, https://docs.apwg.org/reports/apwg trends report q4 2020.pdf

[11] FBI Internet Crime Report 2020, https://www.ic3.gov/Media/PDF/AnnualReport/2020 IC3Report.pdf

[12] Verizon 2020 Data Breach Investigation Report, https://enterprise.verizon.com/resources/reports/2020-data-breachinvestigations- report.pdf

[13] World Health Organization, Communicating for Health, Cyber Security, https://www.who.int/about/communications/cyber-security

[14] Ye Cao, Weili Han, and Yueran Le, Anti-phishing based on automated individual white-list, Proceedings of the 4th ACM workshop on Digital identity management-DIM 08, pp. 51-60, 2008

[15] M. Sharifi, and S. H. Siadati, A phishing sites blacklist generator, 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 840-843, 2008

[16] N. Abdelhamid, A. Ayesh, and F. Thabtah, Phishing detection based associative classification data mining, Expert Systems with Applications, vol. 41, no.13, pp. 5948-5959, 2014

[17] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, Detection of phishing webpages based on visual similarity, Special interest tracks and posters of the 14th international conference on World Wide Web- WWW 05, pp. 1060-1061, 2005

[18] C. L. Tan, K. L. Chiew et al., Phishing website detection using url assisted brand name weighting system, 2014 International Symposium on Intelligent Signal Processing and Communication Systems(ISPACS), IEEE, pp. 054-059, 2014

[19] K. L. Chiew, E. H. Chang, W. K. Tiong et al., Utilisation of website logo for phishing detection, Computers & Security, vol. 54, pp. 16-26, 2015

[20] K. M. kumar, K. Alekhya, Detecting phishing websites using fuzzy logic, International Journal of Advanced Research in Computer Engineering Technology(IJARCET), vol. 5, no. 10, 2016

[21] Rishikesh Mahajan, and Irfan Siddavatam, Phishing website detection using machine learning algorithms, International Journal of Computer Applications(0975-8887), vol. 181, no. 23, 2018

[22] Jitendra Kumar, A. Santhanavijayan, B. Janet, Balaji Rajendran, and Bindhumadhava BS, Phishing website classification and detection using machine learning, International Conference on Computer Communication and Informatics(ICCCI), 2020

[23] Mehmet Korkmaz, Ozgur Koray Sahingoz, Banu Diri, Detection of phishing websites by using machine learning-based URL analysis, 11nth International Conference on Computing, Communication and Networking Technologies(ICCCNT), 2020

[24] Mohammad Nazmul Alam, Dhiman Sarma et al., Phishing attacks detection using machine learning approach, 3rd International Conference on Smart Systems and Inventive Technology(ICSSIT), 2020

[25] Abdulhamit Subasi, Esraa Molah, Fatin Almkallawi,

Touseef J. Chaudhery, Intelligent phishing website detection using Random Forest classifier, International Conference on Electrical and Computing Technologies and Applications(ICECTA), 2017

[26] Structure of a URL image, https://towardsdatascience.com/phishingdomain-detection-with-ml- 5be9c99293e5

[27] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey, Phishing websites features