

## Learning Model for Phishing Website Detection

A. Suryan, C. Kumar, M. Mehta, R. Juneja and A. Sinha\*

Jaypee Institute of Information Technology, Sector 62-A, Noida, UP, India

### Abstract

Website portal empowered with information technology are of great importance in present scenario. With access to data all around the world, securing our information becomes an issue of topmost priority. Over the decade there have been numerous attacks by phishing websites and people have lost huge resources. Such malicious websites, also known as phishing website, steal information of authenticate users and carry out illegal transactions by misusing the personal information. Phishing website links and associated e-mails are sent to billions of users daily, thereby becoming a big concern for cyber security. In this paper, we address the phishing problem using machine learning approach applied on our proposed model, which uses 30 distinct features for phishing detection. We extracted multiple features from the website link and applied appropriate algorithms to classify the link as legitimate or phishing links.

**Keywords:** Information systems, phishing, machine learning, feature extraction, classification, dimensionality reduction, security.

Received on 02 October 2019, accepted on 18 February 2020, published on 13 March 2020

Copyright © 2020 A. Suryan *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.13-7-2018.163804

### 1. Introduction

Internet has privileged everybody to get connected around the world and is one of the biggest advancements in field of communication technology. With the further advancements, internet also spans across several domains, for instance - business, market and online payments portals, there by greatly reducing human effort and resources. Bank accounts, payments, money transfer and official registration can be easily carried online without delay. In such applications, user data is often stored on cloud database that uses the web-based apps as a key interface to retrieve and manage the user data. This calls for several security related issues to be dealt with, of which website phishing of pivotal importance [4,6]. Phishing attack is an online threat to the users that attempts to steal critical information and personal data through malicious websites. Such attack is performed by mimicking the legitimate websites and steal user data through registration forms, bank account details, chats, messages and further uses the stolen data to fraud users [8]. Common phishing attacks include link manipulation, website forgery and covert redirect. In these attacks the users are made to click on attractive advertisements or are redirected to fraud websites where they unintentionally give the attackers

access to their devices and their personal information. Phishing is considered as one of the major cyber menace that pose as significant security threat in present day world and is responsible for the loss of millions of dollars across the globe [9, 11].

### 2. Background Research

Various researches have been conducted to reveal the phished websites and URLs and prevent the user from giving the access to his personal information to malicious websites. Authors Ram B. Basnet and Andrew H. Sung gave the concept of feature selection so as to reduce the dimensions of the dataset which became an important part in the current research [1]. They laid down the concept of correlation between the attributes so as to use their dependency and keep only optimum features to improve the accuracy and reduce the dimensions of the data set by discarding those which add up to the cost of the model without contributing to its performance. Other researchers have attempted to detect the phishing websites using Lexical Features where all the lexical features from the URLs were extracted which included attributes like domain, length, IP and expiration date and the features were fed to the confidence-weighted model [2].

\*Corresponding author. Email: [mailto:adwitiya@gmail.com](mailto:mailto:adwitiya@gmail.com)

Research by Authors Mao and Jian includes the use of Cascading style sheet (CSS) as the base to rate the website on the resemblance in accordance with the website design and layout and using CSS each web page is rated using weighted page-modules similarity [3]. Author Sadia and Afroz in their research gave an approach called Phish Zoo which used the profiles of the legitimate website's appearances for the detection of the malicious websites [5]. The appearances including the images and the website contents are stored in the database and the websites are matched against the trusted websites and common phishing websites for detection. The images were segmented into objects the SIFT- algorithm for image matching was used beside the content matching algorithm which improved the accuracy. In research [16], the authors proposed an Unsupervised Multi-View Hierarchical Embedding framework to address the problem of event-oriented topic mining in microblog stream with constraint of the only textual aspects. Unsupervised Multi-View Hierarchical Embedding can precisely and efficiently aggregate the incoherent latent topics into ones with salient semantic interpretation under a translation-based hierarchical embedding method.

Short texts have become prevalent format of the information on the Internet these days. Inferring the subject of this type of texts becomes a challenging and critical task for many applications. In this paper [17], a model for short text topic modeling, named Conditional Random Field Regularized Topic Model is discussed which first utilizes the Embedding-based Minimum Average Distance to aggregate short texts into regular-sized pseudo documents, which is a generalized solution to alleviate the sparsity problem. Next, the model incorporates global and local semantic correlations by using a Conditional Random Field model to encourage semantically related words to share the same topic label.

Author Jun Hu introduced a method for detection of such phishing websites using the information and references from server logs [10]. When the user visits the phishing website, this website will refer to the legitimate website and ask for resource. Then a log will be identified indicating a phishing URL. Further, the authors Choon Lin Tan and Kang Leng Chiew in their research detected phishing websites by assigning weights to the words in the URL based on their co-appearance at hostname, path and filename of URLs [7]. The words with greater weight were then sent to Yahoo search to get the domain identity with maximum frequency. Then with WHOIS, a query domain name owner was compared to domain name owner of selected domain name. Its accuracy comes out to be 98%. Following the Strategy of LDA model, by using an association matrix to measure the association between latent topics, the authors of [18] have developed an associated topic model, in which consecutive sentences are considered important and the topic assignments for words are jointly determined by the association matrix and the sentence level topic distributions, instead of the document-specific topic distributions only. This approach has given a more realistic modelling of latent topic connections.

Various Clustering techniques have also been applied to the datasets and online reputation services have been used to categorize the data and the information returned is used to rank the data and classify it as phishing or legitimate. Xun Dong gave an approach for detection of the phishing website by analyzing the behavior of the user's [13]. Since the information regarding the sites visited by the user or the information submitted cannot be easily manipulated by the malicious sites this became the basis for his detection method. The monitor gathers the data which the user tried to submit and the detection mode starts where user gets alarmed if the website is malicious which is detected if any violation is found when the data is sent to the website.

The paper [14] has laid down purpose-based access control policies with boundaries and obligations in distributed computing surroundings. The authors have researched the access control framework and also the structure of access policies considering subjects, access actions, purposes, resources and obligations. Algorithms have been designed to help a system in detecting and solving the problems. Authors in paper [15], suggested a unified Branch & Bound (B&B) framework for a class of sequencing problems. The B&B is optimized by designed dominance checkers and caching strategies. The algorithm suggested in this paper can be used to solve optimization problems in several domains. Many experiments were conducted based on the benchmark instances of these problems. The experimentation results were in line with the predicted theoretical results.

In yet another research, the authors proposed a novel method that assembles the domains from the web links which have either direct or indirect association with the malicious site [12]. All the domains collected from the webpage that are directly associated are compared with domains collected from the webpage associated indirectly to reach at a target domain set. On applying Target Identification algorithm, third-party DNS check of the suspicious domains and the targeted domain and after comparison the website was identified as phishing or legitimate. In the research conducted in [19], suggested an approach to predict the next web page to be accessed by the users from their browsing behaviors using clustering and 2nd order Markov model techniques. The prediction of web pages to be browsed by users has many applications in the field of web application development. But this prediction also poses a threat to user's privacy as these algorithms may suggest a user to open a malicious web page which can try to phish user's private data. Therefore, to tackle this issue, we proposed a method to safeguard user's data by detecting legitimate and phishing websites. Thus, increasing the application of the techniques proposed in paper [19] for web applications.

### 3. Dataset Source

The dataset is obtained from the machine learning repository made available by UCL. Phishing Websites Dataset and belongs to the area of Computer Security. The dataset

contains 11055 records with 31 attributes for each record. The first 30 attributes are the actual features of a URL which are provided to the classification algorithms for training. The last attribute is the result attribute which has two values 1 and -1, 1 indicates that the URL in question is legitimate and -1 signifies that the given web link is a phishing URL. The dataset does not contain any missing or N/A values. 0 value for any attribute indicates a suspicion value for that attribute. The other data which was used was some sample phishing URL's taken from the Phishtank Database for final testing and prediction purposes.

## 4. Proposed Framework for Phishing Detection

In the sections mentioned below the complete process of the feature building of URL and dimensionality reduction employed to design our proposed model for detecting phishing. The complete process of model building for phishing detection is described in upcoming subsections.

### 4.1. Feature Building

Every instance in the dataset is represented by the 30 attributes which are mentioned below, these attributes were used to train the classification algorithms mentioned in the next section.

#### URL containing IP Addresses

The existence of IP address at the place of domain name indicates a phishing URL.

$$\Omega_{url,i} |_{i=IP\ address} = \begin{cases} 1 & \text{contains IP in place of domain name} \\ -1 & \text{does not contain IP} \end{cases} \quad (1)$$

#### URL with relatively longer length

Long URL's can be used to cover-up the sceptical part in the URL address. From the dataset an average length of the URL was calculated and it estimated that if the length is less than 54 then it maybe a genuine URL. If the total length lies between 54 and 75 then it may be suspicious else it maybe phishing.

$$\Omega_{url,i} |_{i=url\_length} = \begin{cases} 1 & \text{length} < 54 \\ 0 & 54 < \text{length} < 75 \\ -1 & \text{length} > 75 \end{cases} \quad (2)$$

#### URL with relatively shorter length

Short URL's are used to lead us to the required webpage which is done using HTTP Redirect on a domain redirecting it to the main long URL.

$$\Omega_{url,i} |_{i=short\_length} = \begin{cases} 1 & \text{not tiny url} \\ -1 & \text{tiny url} \end{cases} \quad (3)$$

#### URL containing '@'

If there's a '@' symbol in URLs, it causes the server to not see anything before the '@' symbol.

$$\Omega_{url,i} |_{i=@\ symbol} = \begin{cases} 1 & \text{does not have @} \\ -1 & \text{does have @} \end{cases} \quad (4)$$

#### URL's containing “//” after HTTP/HTTPS Token

The symbol ‘//’ exists in every URL but only once in either the sixth position (HTTP) or seventh position (HTTPS). If anywhere else, it is not legitimate.

$$\Omega_{url,i} |_{i=double\ slash} = \begin{cases} 1 & \text{position of last occurrence double slash} \leq 7 \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

#### URL's containing prefix/suffix separated by '-'

The '-' symbol is very rarely used in the URL but phishing websites do use '-' to confuse user to accept it as a legitimate website.

$$\Omega_{url,i} |_{i=hyphen\ symbol} = \begin{cases} 1 & \text{does not have hyphen symbol} \\ -1 & \text{otherwise} \end{cases} \quad (6)$$

#### URL's containing '.' more than a certain amount

URL might contain dots: '.' multiple times in it. But the count shouldn't be more than certain value keeping in mind the top level, second-level domain and actual name of domain with dot after 'www'. If the remaining dots are more than 1, the URL is suspicious and more than 2, URL is classified as phishing.

$$\Omega_{url,i} |_{i=full-stop\ symbol} = \begin{cases} 1 & \text{symbols in domain part} = 1 \\ 0 & \text{symbols in domain part} = 2 \\ -1 & \text{otherwise} \end{cases} \quad (7)$$

#### HTTPS

HTTPS certificate is essential but in addition to its existence, issuer and age also plays an important role.

$$\Omega_{url,i} |_{i=https\_certificate} = \begin{cases} 1 & \text{Use https} \\ & \text{Issuer Is Trusted} \\ & \text{certificate's age} \geq 1\ \text{year} \\ 0 & \text{Using https} \\ & \text{Issuer Is Not Trusted} \\ -1 & \text{otherwise} \end{cases} \quad (8)$$

#### URL's domain registered time

Phishing websites do not stand long and are have short life. In our program, we find such websites with active use of less than a year.

$$\Omega_{url,i} |_{i=domain\_expiry} = \begin{cases} 1 & \text{Domains Expires on } > 1 \text{ years} \\ -1 & \text{Domains Expires on } \leq 1 \text{ years} \end{cases} \quad (9)$$

### Websites favicon

A favicon is an icon related with the website.

$$\Omega_{url,i} |_{i=website\_favicon} = \begin{cases} 1 & \text{Favicon from same Domain} \\ -1 & \text{Favicon from any external Domain} \end{cases} \quad (10)$$

### Ports

Only those ports are kept open which are under use. Phishers can be a threat if all the ports are opened and they can gain access to information.

$$\Omega_{url,i} |_{i=Port} = \begin{cases} 1 & \text{Port status is of not preferred} \\ -1 & \text{Port status is preferred.} \end{cases} \quad (11)$$

### URL's containing Token 'HTTPS' in domain part

Phishers may add token 'HTTPS' in URL to confuse the user to think of the website as a legitimate one.

$$\Omega_{url,i} |_{i=https} = \begin{cases} 1 & \text{if contains http instead of https} \\ -1 & \text{if contains http} \end{cases} \quad (12)$$

### Websites having links linking to another URLs

This states the fact of the images/videos/links used in the website are from the same domain. Keeping in mind a few of them might be from an external source, we decided about legitimacy of website.

$$\Omega_{url,i} |_{i=links\_percentage} = \begin{cases} 1 & \text{if link percentage } < 22\% \\ 0 & \text{if } 22 < \text{link percentage } < 61\% \\ -1 & \text{otherwise} \end{cases} \quad (13)$$

### Websites having anchor tag linking to another URLs

An anchor is <a> tag. However, we examined that if <a> tag has different domain name as compared to the domain name of the URL.

$$\Omega_{url,i} |_{i=href\_percentage} = \begin{cases} 1 & \text{if href percentage } < 17\% \\ 0 & \text{if } 17 < \text{href percentage } < 81\% \\ -1 & \text{otherwise} \end{cases} \quad (14)$$

### Links in meta, script and link tags linking to other URLs

Usage of meta, script and link tags to offer metadata about webpage, run client-side JS and fetch other resources respectively gives a hint that the website is legitimate. We again check if they are using the same domain name.

$$\Omega_{url,i} |_{i=meta,script,link\ tags} = \begin{cases} 1 & \%age\ of\ meta,script,link\ tags < 17 \\ 0 & 17 < meta,script,link\ tags\ \%age < 81 \\ -1 & otherwise \end{cases} \quad (15)$$

### Server-Side Form Handling

Server Form Handler (SFH) might be having an empty string that might be suspicious as an action is taken on submission. Also, we check for the domain name in SFHs.

$$\Omega_{url,i} |_{i=SFH} = \begin{cases} 1 & \text{SFH not empty or about: blank} \\ 0 & \text{SFH points to other domain} \\ -1 & otherwise \end{cases} \quad (16)$$

### Website requiring user to submit information

Websites have form that can be altered and the information could be misused. This can be don't via using 'mailto:' from client side as well as 'mail' function in server side (PHP).

$$\Omega_{url,i} |_{i=forms} = \begin{cases} 1 & \text{If form not altered with mail function} \\ -1 & \text{If form altered with mail function} \end{cases} \quad (17)$$

### Website's URL should have an identity in WHOIS database

We can fetch identity of a website from WHOIS. For a website's URL to be legitimate, the return value will contain an identity.

$$\Omega_{url,i} |_{i=identity} = \begin{cases} 1 & \text{if has a valid identity in WHOIS database} \\ -1 & Otherwise \end{cases} \quad (18)$$

### Website might be causing redirects

A normal website is redirected one time maximum. Rest if the count is greater it might be suspicious or phishing.

$$\Omega_{url,i} |_{i=redirects} = \begin{cases} 1 & \text{Number of redirects } \leq 1 \\ 0 & 2 \leq \text{Number of redirects } < 4 \\ -1 & Otherwise \end{cases} \quad (19)$$

### Custom URLs

Phishing websites may use JS to not display the real URL. For this, we need source code and check for any 'onMouseOver' event that might be causing any such activity.

$$\Omega_{url,i} |_{i=custom\ urls} = \begin{cases} 1 & \text{No change in status bar on mouse action} \\ -1 & \text{Changes in status bar on mouse action} \end{cases} \quad (20)$$

### Altering with mouse pointer click

Phishers can disable the feature of user able to see the source code. We can detect by checking for an event button which if found causes this.

$$\Omega_{url,i|i=mouse\ click} = \begin{cases} 1 & \text{Right click enabled} \\ -1 & \text{Right click disabled} \end{cases} \quad (21)$$

### Websites containing pop-ups

Although some legitimate websites do use such pop-ups. But majorly, phishing websites use such pop-ups to extract user's personal information.

$$\Omega_{url,i|i=pop-ups} = \begin{cases} 1 & \text{Pop-up window without text fields} \\ -1 & \text{Pop-up window with text fields} \end{cases} \quad (22)$$

### Websites having Iframe used

Iframe can be used to display any additional webpage and can merge it into the original website. It can be also be made invisible and still be working underneath.

$$\Omega_{url,i|i=iframe} = \begin{cases} 1 & \text{Not using Iframe} \\ -1 & \text{Using Iframe} \end{cases} \quad (23)$$

### URL's domain age

This feature is determined from the return values of WHOIS database. We check the age of domain and for phishing websites, it is usually very less.

$$\Omega_{url,i|i=domain\ age} = \begin{cases} 1 & \text{Age Of Domain} \geq 6 \text{ months} \\ -1 & \text{Otherwise} \end{cases} \quad (24)$$

### DNS Record

For legitimate websites, there should be a valid identity coming from data of WHOIS database and there are records for hostname.

$$\Omega_{url,i|i=DNS} = \begin{cases} 1 & \text{DNS record for domain} \\ -1 & \text{No DNS record for domain} \end{cases} \quad (25)$$

### Website customers visits

Website are measured on their popularity which is further measured on the basis of visits. If there are very less visits, then it is a phishing website.

$$\Omega_{url,i|i=rank} = \begin{cases} 1 & \text{Website Rank} < 100,000 \\ 0 & \text{Website Rank} > 100,000 \\ -1 & \text{otherwise} \end{cases} \quad (26)$$

### PageRank

PageRank is a value that starts from "0" and goes till "1". PageRank shows the importance of website on internet. We

found that majority of legitimate websites have PageRank. Rest the phishing websites have a threshold value of PageRank, i.e. "2".

$$\Omega_{url,i|i=pagerank} = \begin{cases} 1 & \text{PageRank} > 0.2 \\ -1 & \text{PageRank} < 0.2 \end{cases} \quad (27)$$

### Google Index

Google's index is done on a website. If a website is indexed by google and is displayed in its search. Usually phishing websites are short-lived and thus, not indexed by google.

$$\Omega_{url,i|i=google\ index} = \begin{cases} 1 & \text{Webpage Indexed by Google} \\ -1 & \text{Otherwise} \end{cases} \quad (28)$$

### Number of Links Pointing to Page

If a webpage is legitimate, there are links pointing to it, either it be of same domain. Since phishing websites are short lived, they don't have any links redirecting to them.

Rule: IF

$$\Omega_{url,i|i=links\ pointing\ to\ the\ page} = \begin{cases} 1 & \text{Number of links pointing to webpage} > 2 \\ 0 & 0 < \text{Number of links pointing to webpage} \leq 2 \\ -1 & \text{Number of links pointing to webpage} = 0 \end{cases} \quad (29)$$

### Statistical-Reports Based Feature

Different companies circulate reports on phishing websites time to time. We will be using two forms: 'Top 10 Domains' and 'Top 10 IPs'.

$$\Omega_{url,i|i=stats-report} = \begin{cases} 1 & \text{Host doesn't belong to Top Phishing Domains} \\ -1 & \text{Host belongs to Top Phishing Domains} \end{cases} \quad (30)$$

The values associated with the features mentioned above were extracted from the URL's to be classified as legitimate or phishing, using the Python programming language, as it comes with comprehensive set of tools and libraries to extract the mentioned features from the URL. The next step of the process is to apply various classification algorithms on the dataset and find the best algorithms having the highest accuracy and then applying dimensionality reduction on the dataset and retrain the models using the best classification algorithms.

For classifying URL's into phishing or legitimate the machine learning algorithms mentioned in Section 4.2 were used. The dataset that was taken from the UCL Machine Learning repository was split into two sets of 70% and 30% and was labelled as training set and testing set respectively.

The algorithms mentioned in Section 4.2.1 to Section 4.2.3 are used to train classifiers using the training set. Cross-validation was used at the time of training with the k-fold set to the default value of 10. Cross-validation ensures

that the testing data is further split into several sets of train and test data and as we used the value of k-fold as 10, the training data was split into 10 such smaller sets, and then these smaller sets of train and test data are fed to the actual algorithm for model building purposes, this ensures that the classifier has been trained with many variations of data before making the final predictions on actual testing data, it also gives the best model with the highest accuracy possible. After this, the accuracy of these classifiers was tested using the testing set. After this analysis it was observed that the Random Forest and Generalized Linear Model (GLM) classifiers gave the best accuracy for the testing data.

The next machine learning algorithm used was Generalized Additive Model (GAM) which is mentioned in Section 4.2.4. This algorithm was used when stacking of two models is used. Model stacking was applied by splitting the original dataset into three parts of 48%, 32% and 20% labelled as training, testing and validation sets respectively. The training set was fed to the Random Forest and GLM algorithms with a k-fold value of 10. Then these two classifiers were used to predict the results for the testing set. The predictions from both the classifiers on the testing set were stored in a data frame as separate columns, with a separate column for the original classes of the testing set for every instance in the testing set. This data frame is labelled as new training set. The validation set is also used to predict results from the two trained classifiers and a similar data frame is built using the prediction from the two classifiers on the validation set and the original result column of the validation set. This data frame built from the validation set is labelled as new testing set. The new testing set is fed to the GAM algorithm to train a classifier and the new testing set is used for testing the model and after testing the accuracy of this model was also noted. The applied technique is referred to as stacking of models.

## 4.2. Classification Techniques Used

Following section provides results from our experimentation, conducted with machine learning approach.

### Random Forest

Random Forest (RF) is an extension to the Decision Tree classification algorithm. It is an ensemble learning method for classification or regression. Random Forest works by creating multiple decision trees for the given set of training data (default number of tree's are 100). For predicting the class for a sample input that input is provided to each of the decision trees created in this algorithm, each tree returns an output class for the given sample and the output class which is returned maximum number of times is returned by the Random Forest algorithm and is the final class for the given input sample.

In the figure 1. the X-Axis represents the  $M_{try}$  value for the Random forest, which decides the number of randomly chosen predictors at each split for building a tree. Y-Axis

represents the Out-of-bag (OOB)error. The OOB error comes out to be minimum for an  $M_{try}$  value of 10.

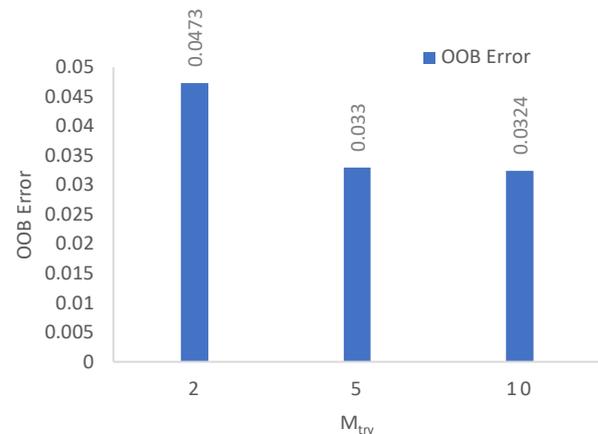


Figure 1. OOB Error for different  $M_{try}$  value

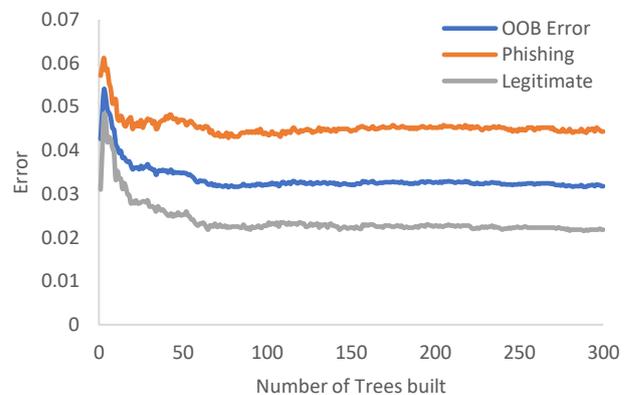


Figure 2. Random Forest Error Values

In the above figure 2, the X-Axis represents the number of trees built in Random Forest. The green plot on the Y-Axis represents the OOB error for different number of trees, the black plot represents the root-mean squared error in class 1 for different number of trees and the red plot represents the root-mean squared error in class -1 for different number of trees. All the errors stop decreasing till the number of trees reaches to 300.

### Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm used for data classification purposes or regression. SVM works on the idea of finding hyperplanes that divide the dataset into their respective classes. Support vectors are actual dataset points which are closest to the hyperplanes and are considered as critical points in the dataset and if these points are removed then the hyperplanes dividing the dataset will also be shifted.

### Generalized Linear Model

The Generalized Linear Model classification algorithm is a flexible generalized linear regression algorithm which relates the linear models to the response variables through a

function defined as link function. The link function allows the magnitude of variance of every measurement to be a function of its predicted value.

### Generalized Additive Model

Generalized Additive Model is a very useful algorithm which can be used to train models when holdout stacking of models is applied. In this paper we used the stacking or combining of the Random Forest and GLM algorithms to produce the best results of accuracy for all pairs of classification models used.

### Recursive Partitioning and Regression Trees

Recursive partitioning is a statistical method for multivariable analysis. Recursive partitioning is used to create a decision tree that strives to correctly classify members of the population by splitting it into sub-populations based on several independent variables. The implementation of this algorithm in R i.e. the classification and regression trees (CART) algorithm is called Recursive Partitioning and Regression Trees (RPART).

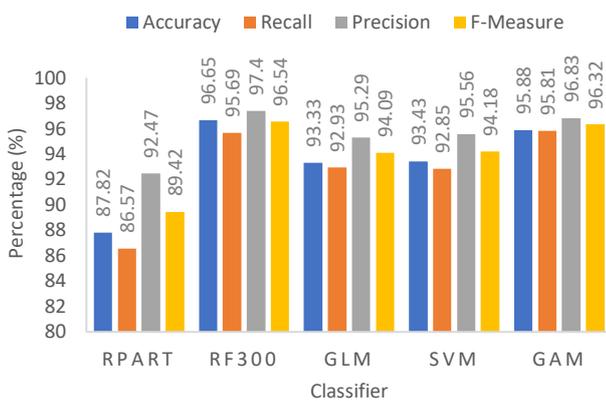


Figure 3. Classifiers Performance Metrics

The above figure 3. shows the accuracy, recall, precision and f-measure values for all the classification models used before applying dimensionality reduction. RPART is the simple decision tree classifier, RF300 is the Random Forest classifier with 300 trees, GLM, SVM, and GAM are mentioned in Section 4.2.

The next step was to apply dimensionality reduction. For this, Principal Component Analysis (PCA) was used which is mentioned in Section 4.3. To reduce the number of attributes and improve the accuracy for the given dataset we first find the correlation between each pair of attributes. The pairs having correlation of 0.8 or more were selected and for each pair of attributes the PCA values were calculated. It was observed that the PC1 values for each pair were greater than the PC2 values, PC1 and PC2 being the Principal Components for the pair of attributes. Each of the selected pair of attributes were combined to form new attributes by multiplying the PC1 values with the attribute value in that pair and then summing up these multiplied values for each attribute in the pair. After adding the new attributes, the old selected pairs of attributes were removed from the dataset.

This new formed dataset was split into 70% and 30% sets labelled as testing and training. The Random Forest algorithm was used for training of this new dataset as this algorithm gave the best accuracy on the original dataset. The trained model was tested using the testing set and the accuracy was observed to be better than the other models trained so far. The following step was to use the PCA trained model to classify a URL as legitimate or phishing. For this, the sample URL was passed to a program written in Python programming language and the values for each of the 30 attributes mentioned in Section 4.1 were extracted from the URL and were stored in the form of a vector. Then this vector was passed to the PCA trained model for classification. The final class of the URL was displayed to the user. 1 being the URL was legitimate and -1 being that it was phishing. This was done for several test URL's taken from the Phishtank database and some very commonly used URLs.

### 4.3. Dimensionality Reduction Techniques

With the increase in the amount of data these days, dimensionality reduction has become the need of the hour. Dimension reduction is a method to reduce the size of data by extracting relevant information and disposing rest of data as noise. With the increase in the number of attributes in a given dataset, the outcome accuracy of the machine learning algorithm also increases and the chances of better prediction also improves. But some unnecessary attributes might lead to inconsistent predictions which results in overall decrease in accuracy of the algorithm applied. Dimensionality reduction is used to remove or modify these unnecessary attributes. Principal Component Analysis (PCA) is one such technique to deal with the problem and remove inconsistent results.

Principal Component Analysis (PCA) is a very popular linear dimension reduction algorithm. It may be used alone or may be used as a starting point for other dimensionality reduction algorithms. The algorithm works by finding principal components for every attribute in our dataset and gives us an idea about the importance of different pairs of attributes in the prediction of accurate results.

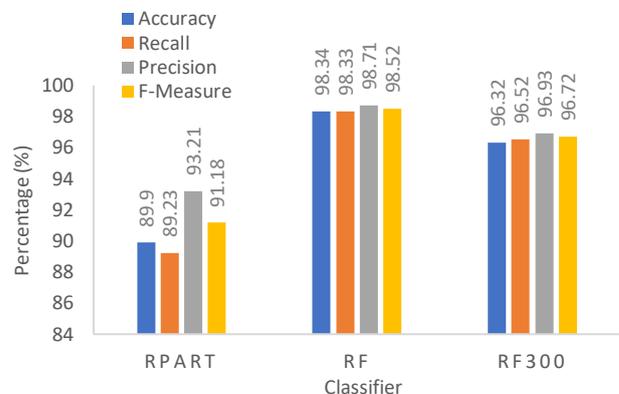


Figure 4. Classifier Performance Metrics after Dimensionality Reduction

The following figure 4. shows the accuracy, recall, precision and f-measure values for all the classification models used, after applying dimensionality reduction using principal component analysis.

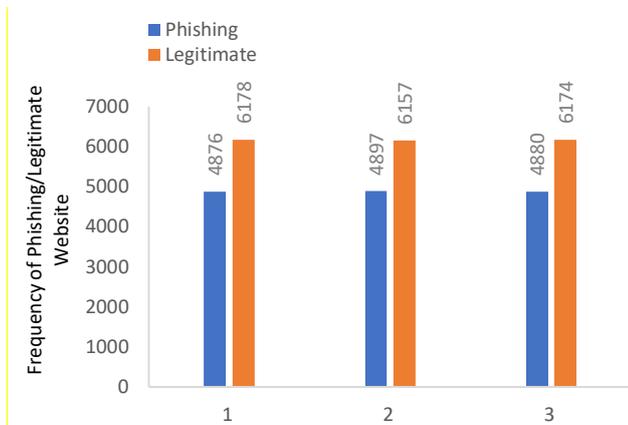


Figure 5. Frequency Plot of Phishing and Legitimate URLs

Figure 5. shows the frequency of phishing and legitimate URLs in the original dataset (1), predicted after classification (2) and predicted after dimensionality reduction (3). The machine learning models, after dimensionality reduction, were tested against real time data and the following results were predicted where 1 indicates legitimate website and -1 indicates phishing website.

Table 1. Prediction Result for Real Time URLs

URLs	Actual	Predicted
https://ah8kbfikwjuhtve9rafpma-on.driv.tw/	-1	-1
https://accounts.apple.comss.live/	-1	-1
https://itacard-mobi-promocao.com/nhh	-1	-1
https://login.bog.ge/ibank/	-1	-1
https://www.audible.com/?ref=Adbl_ip_rdr_from_DE&source_code=AUDGBHP09	-1	1
https://www.the-rio.com/nexi/3AmE9auKokXej4IwdvWbspXQFLiq0HDBPzOySZUt5	-1	-1
https://www.the-rio.com/nexi/3AmE9auKokXej4IwdvWbspXQFLiq0HDBPzOySZUt5	-1	-1
https://www.facebook.com	1	1
https://www.youtube.com	1	1
https://www.amazon.in/	1	1
https://whatsapp-my.maulanainfo.xyz	-1	-1
https://www.phishtank.com/phish_search.php?valid=y&active=y&Search=Search	1	1
https://sarahkurier.com/oplata24/	-1	1
https://www.flipkart.com/order_details?order_id=OD117570157595212000&src=od&link=track	1	1
https://secure.amazon.com.update.fkubl.com/safety/92e9181db6cd439082d899b295b0b8ff/	-1	-1
http://oath-yahoo-auth0.duckdns.org/login/login.yahoo.com.php?cmd=login_submit&id=10ff0b5e85e5b85cc3095d431d8c08b410ff0b5e85e5b85cc3095d431d8c08b4&session=10ff0b5e85e5b85cc3095d431d8c08b410ff0b5e85e5b85cc3095d431d8c08b4	-1	-1
http://zonasegura.viabcp-pe.com/iniciar-sesion	-1	-1

Hence, above results justify the use of machine learning models to our approach that considered 30 features to perform prediction with 88.24% accuracy, in reference to the above set of URLs.

### 5. Conclusion

With the increase in usage of social media and online services like e-payment services, shopping portals, online commercial outlets, etc., the amount of data publicly available is tremendous and is easily accessible. Due to this reason phishing has become one of the biggest cyber security threats of the century. In our research, we have tried to tackle the problem by classifying an URL as phishing or legitimate by applying machine learning to our proposed model that considers 30 distinct features for efficient classification. Further, predictions are enhanced by applying the concept of dimensionality reduction. The experimentation result shows an accuracy of 90% over a set of websites links. As future research direction, the accuracy of our predictions can be improved by applying neural networks and deep learning algorithms, along with application of higher-order dimensionality reduction techniques, for instance Variance Inflation Factor (VIF).

### References

- [1] Ram B. Basnet, Andrew H. Sung, and Quingzhong Liu. "Feature selection for improved phishing detection". In: Jiang H., Ding W., Ali M., Wu X. (eds). *Advanced Research in Applied Artificial Intelligence. Proceedings of the 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2012, June 9-12, 2012; Dalian, China, Heidelberg: Springer; 2012. pp. 252-261.*
- [2] Blum, Aaron, Brad Wardman, Tamar Solorio, and Gary Warner. "Lexical feature based phishing URL detection using online learning". *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security; October, 2010; Chicago, Illinois, USA. ACM; 2010. pp. 54-60.*
- [3] Jian Mao, Wenqian Tian, Pei Li, Tao Wei, and Zhenkai Liang. "Phishing-alarm: robust and efficient phishing detection via page component similarity". *IEEE Access. 2017; 5(2017): pp. 17020-17030.*
- [4] Jian Mao, Jingdong Bian, Wenqian Tian, Shishi Zhu, Tao Wei, Aili Li, and Zhenkai Liang. "Phishing page detection via learning classifiers from page layout feature". *J Wireless Com Network. 2019; 2019(1): p. 43.*
- [5] S. Afroz and R. Greenstadt, "PhishZoo: Detecting Phishing Websites by Looking at Them". *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing; September 8-21, 2011; Palo Alto, CA, USA: IEEE; 2011. pp. 368-375.*
- [6] M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah, "Predicting Phishing Websites Using Classification Mining Techniques with Experimental Case Studies". *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations; April 12-14, 2010; Las Vegas, NV, USA: IEEE; 2010. pp. 176-181.*
- [7] C. L. Tan, K. L. Chiew and S. N. Sze, "Phishing website detection using URL-assisted brand name weighting system".

- Proceedings of the 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS); December 1-4, 2014; Kuching, Malaysia: IEEE; 2014. pp. 054-059.
- [8] M. N. Feroz and S. Mengel, "Phishing URL Detection Using URL Ranking". Proceedings of the 2015 IEEE International Congress on Big Data; June 27 – July 2, 2015; New York, NY, USA: IEEE; 2015. pp. 635-638.
- [9] A. Y. Daeef, R. B. Ahmad, Y. Yacob and N. Y. Phing, "Wide scope and fast websites phishing detection using URLs lexical features". Proceedings of the 2016 3rd International Conference on Electronic Design (ICED); August 11-12, 2016; Phuket, Thailand: IEEE; 2016. pp. 410-415.
- [10] J. Hu et al., "Detecting Phishing Websites Based on the Study of the Financial Industry Webserver Logs". Proceedings of the 2016 3rd International Conference on Information Science and Control Engineering (ICISCE); July 8-10, 2016; Beijing, China: IEEE; 2016. pp. 325-328.
- [11] A. J. Park, R. N. Quadari and H. H. Tsang, "Phishing website detection framework through web scraping and data mining". Proceedings of the 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON); October 3-5, 2017; Vancouver, BC, Canada: IEEE; 2017. pp. 680-684.
- [12] Ramesh, Gowtham, Ilango Krishnamurthi, and K. Sampath Sree Kumar. "An efficacious method for detecting phishing webpages through target domain identification". Decision Support Systems. 2014; 61 (2014): pp. 12-22.
- [13] Dong, X., Clark, J.A. & Jacob, J.L. "Defending the weakest link: phishing websites detection by analysing user behaviours". Telecommun Syst. 2010; 45: pp. 215–226.
- [14] Wang, Hua, Lili Sun, and Elisa Bertino. "Building access control policy model for privacy preserving and testing policy conflicting problems". Journal of Computer and System Sciences. 2014; 80(8): pp. 1493-1503.
- [15] Z. Zhang, L. Teng, M. Zhou, J. Wang and H. Wang. "Enhanced Branch-and-Bound Framework for a Class of Sequencing Problems". IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2019; pp. 1-11.
- [16] Peng, M., Zhu, J., Wang, H., Li, X., Zhang, Y., Zhang, X., & Tian, G. (2018). "Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding". ACM (TKDD). 2018; 12(3): pp. 1-26.
- [17] Gao, W., Peng, M., Wang, H., Zhang, Y., Xie, Q., and Tian, G. "Incorporating word embeddings into topic modeling of short text". Knowl Inf Syst. 2018; 61(2): pp. 1123-1145.
- [18] Jiang, H., Zhou, R., Zhang, L., Wang, H., & Zhang, Y. "Sentence level topic models for associated topics extraction". World Wide Web. 2018; 22(6): pp. 2545-2560.
- [19] Khalil, F., Wang, H., & Li, J. "Integrating markov model with clustering for predicting web page accesses". Proceeding of the 13th Australasian world wide web conference; June 30 – July 4, 2007; Coffs Harbour, Australia: AusWeb; 2007. pp. 63-74.