MACHINE LEARNING APPROACH FOR REAL TIME BASED PHISHING WEBSITE DETECTION SYSTEM

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

B.Tech.

by

Lekhani Agrawal (2014IPG-051)
Shivani Mandeliya (2014IPG-083)
Vaishnav Chandak (2014IPG-127)

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR-474 010

2017

CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, entitled **MACHINE LEARNING APPROACH FOR REAL TIME BASED PHISHING WEBSITE DETECTION SYSTEM**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *May 2017* to *October 2017* under the supervision of **Dr. Ajay Kumar**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisors

ABSTRACT

Over the past years, there has been an increase in the amount of phishing attacks. Phishing is a malicious practice in which the attacker fraudulently acquires confidential information like bank details, credit card details, or passwords from legitimate users. It is a spurious approach that aggregately uses technology and social engineering in order to obtain personal and sensitive user information. In phishing, users are tricked with an formal looking mail or messages that are made to look like they had come from legitimate sources like ecommerce sites, financial institutions, etc. Here we propose an anti-phishing technique to safeguard our web experiences. Our approach uses the Lexical features, Host based features and site popularity features of a website to detect any suspicious or phishing website. These features are obtained from the source code by taking URL as input and then these features are fed to the classifier algorithm. The results obtained from our experiment shows that our proposed methodology is very effectual for preventing such attacks as it has 95.26% accuracy.

Keywords: Social Engineering, URL, Lexical features, Host-based features,Phishing, SVM,KNN,ANN,Logistic-Regression,Gradient Boosting,Tree-Bagging Classifier,Decision-Tree Classifier.

ACKNOWLEDGEMENTS

We express our gratitude to **Dr. Ajay Kumar** for guiding us and for giving us the opportunity as well as the autonomy to experiment with our own ideas. We are obliged to him for not only his academic guidance but also for the personal efforts that he made to motivate and energize us. We would like to thank him for the constant faith and support that helped in nurturing and blossoming of the present project. His valuable guidance, suggestions, criticism and the judgment helped in exceling in the domain that we chose for this project. Our mentor always answered our strangest of the questions and confusions patiently and with smiling graciousness. He always appreciated our views and give us the opportunity for further improvements by taking personal interest in the project. The present stage of our project is the result of patience, motivation, suggestions and the overwhelming support of our mentor.
Finally, we are indebted to our Institute and colleagues who encouraged us to work harder. Their support served to renew our spirit and helped to refocus our attention and energy in order to carry out this work successfully.

(Lekhani Agrawal) (Shivani Mandeliya) (Vaishnav Chandak)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SVM	Support vector machine
KNN	K-nearest neighbours
GUI	Graphical User Interface
URL	Uniform Resource Locator
ANN	Artificial Neural Network
FP	False Positive
FN	False Negative
TP	True Positive
TN	True Negative

NOTATIONS

$B < A$	B is less than A
$B > A$	B is greater than A
$B \leq A$	B is less than or equal to A
$B \geq A$	B is greater than or equal to A
D_K	Set of Feature Vector of Websites
x_{ij}	Value of j th Feature of i th Website
y_i	Label for the Website represented by x_i
$\|\|w\|\|_1$	L1-norm of the weight vector
d(x,y)	Euclidean distance between vector x and vector y
N	Set of Positive Integers
W	Set of Whole Numbers
Z	Set of Integers
Z^+	Set of Positive Integers
Z_n	Set of non-negative integers less than n

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

1.1 General

In this era of internet and digitization, everything is available at the tip of a smartphone. Everyone is using web services for online shopping, business development, studies etc. There are some services where the user need to share some information with the server. Due to these reasons there has been an escalation in the intrusion activities to steal the personal information such as password and credit card information. There has been an increase in phishing activities during this decade hence, threatening the public to switch from offline to online system of activities. These malicious web sites largely promote the growth of Internet criminal activities and constraint the development of Web services. Detection of these phishing websites is a really important safety measure for most of the online platforms. So as to save a platform with malicious requests from such websites, it is important to have a robust phishing detection system in place.

1.2 Background

An identity theft that occurs when a malicious web site masquerades a legitimate one is called Phishing. Such a theft occurs in order to procure sensitive information such as passwords, bank account details, or credit card numbers. It is a kind of deception technique and it exploits social engineering and information technology to capture sensitive and personal information, such as passwords and credit card details by masquerading as someone who is trustworthy or benign business in an electronic communication. This information is then abused illegally to do criminal activities. Phishing makes use of spoofed emails which look exactly like an authentic email. These emails are send to a bulk of users and appear to be coming from legitimate sources like banks, e-commerce sites, payment gateways etc. Such phishing emails attract the users to visit fraudulent websites through links provided in it. The makers of such illegitimate website made them exactly look like a legitimate one so that no user can identify the difference easily. The phishing attackers use different kind of social engineering tactics to lure users for example: giving attractive offers to just visit the site. Some threatening tactics like suspending the account if some (fake) update process is not done are also used to persuade the user to use such website.

1.3 Objective

The purpose of this project is to develop a Real Time Phishing Website Detection System that warns the user whenever the user visits a malicious or fraud website.
According to the Anti-Phishing Working Group, there were 18,480 unique phishing attacks and 9666 unique phishing sites reported in March 2006. Phishing attacks affect millions of internet users and are a huge cost burden for businesses and victims of phishing (Phishing 2006). This system will monitor the Uniform Resource Locator (URL) of every web service that the user is accessing and before any exchange of activities happen, it will warn the user to not go through a website if it is found malicious. In this project, we will be using Machine Learning techniques and will be applying the different models and algorithms to develop an efficient system with the highest possible accuracy. We will also deploy neural network to categorize the different website according to the risks involved (if the user access them).

1.4 Phishing Life Cycle

A fake webpage for the most part contains a login form where client fills his own data(personal) and attacker utilizes this private data for his own monetary profit. Following steps are engaged with a Phishing attack:
Step 1: The attacker duplicates the all content from the site of a notable organization , bank or government and makes a Phishing site. The attacker keeps the visual comparability between the original and the fake one.
Step 2: Assailant sends bulk number of Email to the clients.
Step 3: The client opens the email and visits the phishing site. The phishing site approaches the client for his/her credentials.
Step 4: The assailant catches the individual personal data of the client by means of fake site and uses this data for individual or some money related purpose.

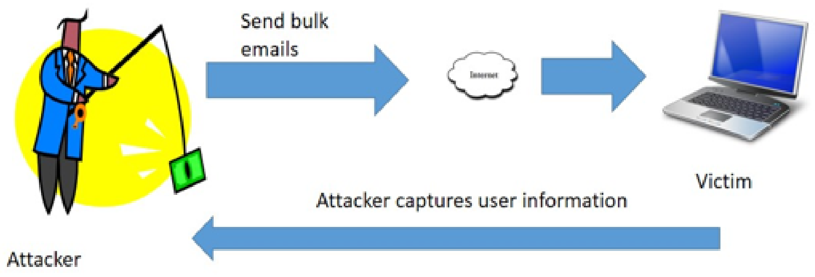


Figure 1.1: Phishing Life Cycle

1.5 Literature Review

Typically phishing attacks can be exploited by sending spoofed link to the trusted user and then redirected them to bogus website. Whenever user will enter the important information to that fake website then immediately this information will store to the system of the hacker and then finally the user will be redirected to any other websites that will not be related to the user .There were many research that were conducted to detect phishing attacks.

Phistank is a website which identifies if a website is legitimate on the basis of the data stored in it. It uses the black list based approach which is fast but has disadvantage of very low detection rate.

Anti-phishing application by A.Fu and Deng provided solution to the problem of recognizing phishing URLs that make use of various HTML content such as Flash ,Pictures and Java Applet. Their approach is based on the visual similarity between suspicious webpages and the benign webpages using the Earth Mover's Distance (EMD).The pages of the websites are first converted to very low resolution images and then sampled and expressed as image features which is made up of dominant color category and the corresponding centroid coordinate for the evaluation of similarity (A. Fu and Deng, 2006).

W.Liu provided an antiphishing approach that uses visual features which analyses the visual similarity between the suspicious site and the already present legitimate website. This method took a variety of visual features into consideration. To identify a malicious website, the method makes use of two modules. First module is runned on the local server to check suspicious URLs and second module checks the visual similarity

among stored legitimate webpage and the suspicious webpage (W Liu, 2006)(Chowdhury and Islam, 2006).
There is a content based approach for phishing-detecting proposed which pays attention particularly to the Google Page Rank. Google page rank is used to do the evaluation of a website but this technique is not reliable always. (Zhang, 2010)
Also, an approach which uses URL feature, which emphasises on phishing URLs which was embedded with legitimate domain names was proposed. They requested a search engine to get returned results as ranks by passing the domain names of websites.Then a rank based classifier was used to differentiate phishing URLs from the legitimate ones. However, in this approach detection was less comprehensive because it targeted only a subset of malicious campaigns where the URLs were embedded using legitimate domain names (M. Khonji and Iraqi, 2011).
There is a one more approach that examines the host-based features as well as the lexical features of a webpage.In this approach query URL are divided into domain, path and other parameters. Through the parameters feature vector for the URL is constructed. It is then passed to a logistic regression classifier with feature vector as input for evaluation of phishing probability. (Zhang and Wang, 2012).
There are Search Engine Based techniques developed for Anti-Phishing. These techniques primarily uses the information provided by search engine and this information is used for the detection purpose.They make use of search engines for detection. This is an entirely new type of approach used for phishing detection. These are actually the login logic based solutions. We know that Phishing websites cannot have access to the user credentials or the user sensitive information. Therefore a phishing website will not be able to tell whether a credential is wrong or correct. This approach exploits this fact and suggests that a website can be tested against wrong username and password pair. It makes use of this method (Huh and Kim, 2012)(Hongl, 2009) (Yue and Wang, 2010)
An approach was proposed which uses the logo present in the webpage rather than executing full page matching. Their approach was based on the logic that logo can be used to verify the identity of a suspicious webpage (Afroz and Greenstadt, 2015).
One the service which the Google provides for safe browsing allows to check the URL against a list of malicious domains that is constanly updated by Google. The Safe Browsing Lookup API permits to pass the suspicious website's url to Safe Browsing service which tells if the URL sent by the client is benign or malicious.The client URLs are verified using the malicious and phishing lists maintained by Google. However,this approach has the following shortcomings: (i) Before sending the URL hashing is not performed and (ii) There isn't any constraint over the response time taken server to lookup (*Google safe browsing API*, 2015)
Adulghani Ali Ahmed proposes an approach where he focuses on the view to interact with the fake websites but not with the real ones. This approach differentiates between

the benign and fake website by considering the URL of only fake urls. But this approach is not efficient as it don't pay attention to the discriminative features as well accuracy or the detection rate (Adulghani Ali Ahmed, 2016)
An approach was proposed to prevent against the Phishing attacks by the use of auto-updating white lists. In this approach, when user tries to open a website, the browser warns him/her to not access the website if that website is not available in the white-list. This technique also examine the legitimacy of given website using hyper link features.Hyperlinks are extracted from the source code of given webpage and are passed as an input to the proposed phishing detection algorithm. This proposed approach is effective as it has true positive rate of 86.02 % while false negative rate lesser 1.48%.(Jain and Gupta, 2016)
In recent days, attackers use software toolkits to create a huge number of phishing website URLs. So,the approach that uses blacklist can not detect phishing websites efficiently (*Statistics about phishing activity and phishtank usage*, 2013) (W Liu, 2006).
There is an another technique that is developed which focuses on detecting phishing attacks but on mobile phones using anti-phishing scheme on mobile based platforms. It was a challenging task to apply this technique because mobile phone and users have more limitation with them (Longfei WC, 2016)
Mona Ghotaish Alkhozae proposed an approach which was based on examining the webpage source code.They extracted few phishing characteristics from the World Wide Web Consortium (W3C) and then evaluate if the websites are secure. They examine every character in the source code to identify a phishing character. If any phishing character is identified, they decrease the initial secure weight. Finally they calculate the security percentage. More the security percentage, more secure is the website. They examined two webpage source codes for benign and malicious websites and compare the security percentages among the two of them (Mona Ghotaish Alkhozae, 2015).
Client side detection for application and browser add on can also be used for the detection of fake or phishing websites.This proposed an approach which focuses on this method. Their method finds the fake website by inspecting the URL of the website. The database of black list and white list is utilized to show fake website by using pattern matching (A.Bavani, 2017)

1.6 GAP Analysis

A.Fu and Deng provided solution method that had the drawback that most image-based anti-phishing approaches are likely to be influenced by a tremendous variety in the appearance of phishing websites.
In content-analysis based approach proposed by J.Zhang in 2010 based on Google page rank newly built up sites Google page rank will give a low value of page rank eventually marking the new website in the category of suspicious websites. However, the website may be legitimate in this case but the approach fails to recognize this fact.
Huh and Kim in 2012 proposed Search Engine Based techniques which had the deficiency in this approach is that what if the search engine fails and deceives in case and the main flaw of this method was the time over head as it is not an easy task to test all the accessed websites so this was not the reliable solution.
Afroz and Greenstadt's logo based approach needed large storage to stockpile the collection of logos taken from legitimate websites and it was very handy task to keep up the logo update pace with the ever increasing pace of newly emerging logos on the internet.
Jain and Gupta as well as A.Bavani proposed to make use of autoupdating white lists and black lists which had the challenge of keeping up with the rate at which websites are being developed today.

CHAPTER 2

DESIGN DETAILS AND IMPLEMENTATION

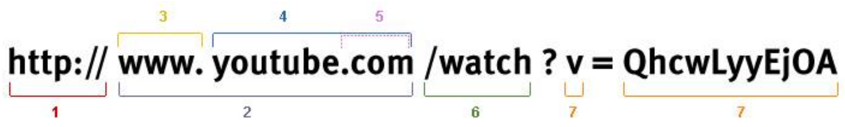
2.1 Design Details

2.1.1 Data

Data is distinct pieces of facts,statistics and information that is collected together and usually formatted in a special way for reference or analysis.
Our mechanism uses the Uniform Resource Locator (URL) itself as data without accessing the content of Web sites and analyzes it.
We have used two types of urls in our dataset which are urls of malicious sites and urls of benign sites.

2.1.1.1 URL

A URL is an acronym for (Uniform Resource Locator) which is basically a specific type of URI (Universal Resource Identifier) which gives a reference to an existing resource on the Internet.
A URL basically consists of several components. To learn the structure and the components of the url , we will use the following example of the URL:



{
1. The Protocol in use in this case: HTTP (Hypertext Transfer Protocol).
There are also other protocols like HTTPS, FTP, MAILTO and so on.

It refers to the name of the protocol to be used to obtain the resource.

2. The Host or Hostname: www.youtube.com.
It refers to the name of the web server on which the resource is available.
It is basically, the “domain” to which the URL is referring.

3. The Subdomain: www.
It is a domain that is a part of a main domain.

4. The domain name (Domain): youtube.com .
IP addresses are determined using Domain names.

5. The Top-Level-Domain (a web-address suffix): .com
Also known by the shorthand TLD it refers to the last segment of the domain name.

6. The Path: ‘/watch’.
A path usually points out to a file or folder (directory) on the machine (for example “/folder/file.html”).

7. Parameter and value: v (Parameter), QhcwLyyEjOA (Parameter value)
Parameters are initialised by the “?” inside the URL.In the given url, “v” is the parameter name and the value of the parameter is “QhcwLyyEjOA” (Name of the parameter and its value always have the same structure: Parametername=Parametervalue)
}

2.1.1.2 Data collection

The URLs of benign webistes were collected from (*The Web information company – www.alexa.com*, 2017) and phishing websites from (*Phishtank: www.phishtank.com*, 2017).
The dataset consists of 1613 malicious urls and 1450 benign urls and then divided the it into two dataset that we are using for training purpose and testing purpose.

- In the train dataset, we have used 1113 malicious urls and 1000 benign urls.
- In the test dataset, we have used 500 malicious urls and 450 benign urls.

2.1.1.3 Feature extraction

Feature extraction is analyzing data and crawling to get relevant information from web pages or data sources in a particular pattern. The data extraction is majorly done from unstructured and different data sources. The unstructured data may be in the form of tables/ indexes. It is a complex process to perform but various open source tools are available to simplify. The motive in this section to show how the features of websites are extracted, which can be used to classify phishing and legitimate website.
We have extracted three categories of features from the URLs which are as follows:

1. Lexical Features:
It is observed that the URLs of many illegal sites look different usually as compared to original and benign websites. These are called lexical features. Analysis of such lexical features offers the opportunity to capture the property for classification purposes.
We first distinguish the following two sections of a URL:
The Host name and the Path, from which we obtain collection-of-words (strings delimited by /, ?, .,=, - and) (Anjali B. Sayamber, 2014). On analysis, we found that phishing website prefers to have longer URL. They contain generally more levels (delimited by dot), have more tokens in domain and path, have longer token. Besides, phishing and malware websites could pretend to be a benign one by containing popular brand names as tokens other than those in second-level domain. Phishing websites and malware websites can contain IP address directly so as to cover the suspicious URL, which is found very rarely in case of legitimate websites. Also, phishing URLs are found to contain several suggestive word tokens (confirm, account, banking, secure, webscr, login, sign in). We have checked the presence of these security sensitive keywords to extract some important characteristics of phishing web pages and then assign binary values to the output in order to further utilize these properties for the training and testing the dataset.

We have used the following lexical features obtained from the url:
{

1.token_count - It counts number of tokens in the URL
2.avg_token_length - It shows the avg length of the token present in the URL .
3.No_of_dots - It shows the number of dots present in the URL.
4.Length_of_url - It shows length of the URL.
5.avg_path_token_length - It shows the average length of token in the path of the URL.
6.avg_domain_token_length - It shows the average length of the token present in the domain of the URL.

7.path_token_count - It counts the number of token in the url path.
8.path - It will give the string of the path of the URL.
9.largest_domain - It shows the length of largest token in the domain.
10.domain_token_count - It counts the number of token in the domain of the URL.
11.largest_path - It shows the length of largest token in the path.
12.largest_token -It shows the length of largest token in the url.
13.sec_sen_word_count -It shows the number of words that are security sensitive.
}

2. Site popularity Features:
Each day, thousands of malicious and malware websites are hosted but still their popularity is less as compared to legitimate websites as they do not have any useful information for the users. They have less traffic. Site popularity features shows how much a web page is popular among Web users. This is one of the methods which is used by Google to evaluate a page importance. Every month, Google does its re-indexing to change the maximum page_rank of all pages on the web. For this reason, we have considered site popularity as an important feature. We have referred this Traffic feature from Alexa.com.
We have used the following Site popularity features obtained from the url:

{
1.rank_host(site) - It gives the rank given to hostname of the website .
2.rank_country(site) - It gives rank given to the website in the country by the search engine.
}

3.Host-based Features:
Host-based features explain "where" phishing sites are hosted, "who" they are managed by, and "how" they are administered. We can use host based features because phishing Websites sometimes hosted in less reputable hosting centers and on the machines those are not usual Web hosts, or through not reputable registrars. These features are based on the fact that malicious sites always registered in less reputable hosting centers.
We have used the following Host-based features obtained from the url:
{
1.ASNno - It represents the autonomous system number of the ISP(Internet Service Provider).
2.IPaddress_presence - It checks the presence of the IP in the URL.
3.Length_of_host - It gives the length of the host of the URL.
4.host - it will give the string of host from the URL string.

}

4.Safe Browsing:
We are also extracting safebrowsing feature from the url .The main aim of the Safe Browsing system is that it checks for the bad URLs that you're browsing. It check URLs against lists of unsafe web resources that Google instantly updated. Browsing protection currently protecting users from deceptive sites, malware sites which includes phishing and social engineering sites and the sites that are hosting potentially unwanted software.We are also extracting safebrowsing feature from the url . The main aim of the Safe Browsing system is that it checks for the bad URLs that you're browsing. It check URLs against lists of unsafe web resources that Google constantly updated.Browsing protection currently protects the users from deceptive sites, malware sites which includes phishing and social engineering sites and the sites that host potentially unwanted software.

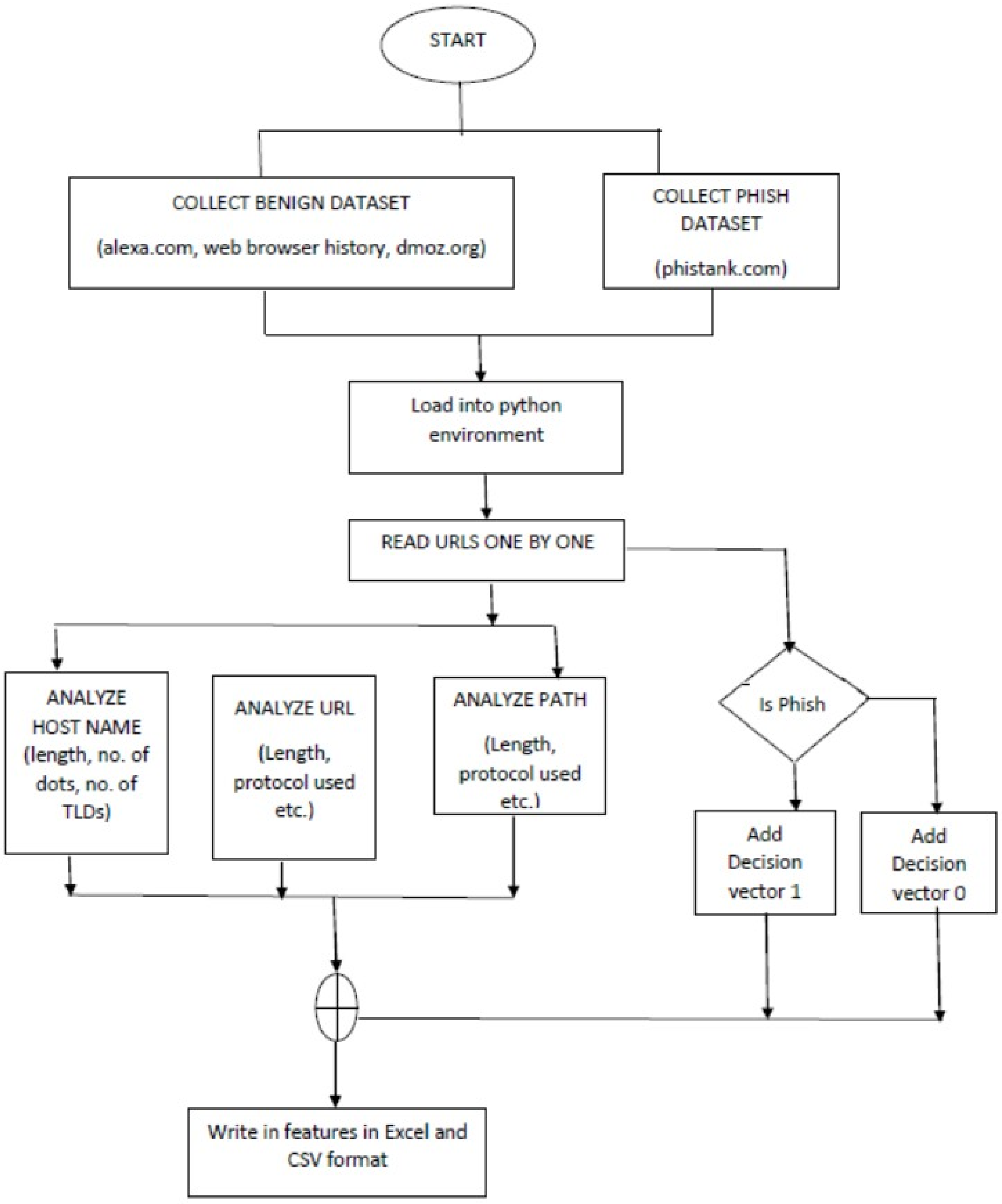

START
COLLECT BENIGN DATASET
(alexa.com, web browser history, dmoz.org)
COLLECT PHISH DATASET
(phistank.com)
Load into python environment
READ URLS ONE BY ONE
ANALYZE HOST NAME (length, no. of dots, no. of TLDs)
ANALYZE URL (Length, protocol used etc.)
ANALYZE PATH (Length, protocol used etc.)
Is Phish
Add Decision vector 1
Add Decision vector 0
Write in features in Excel and CSV format

Figure 2.1: Feature Extraction Flowchart

2.1.1.4 Data Preprocessing and Cleaning

This presents a technique of data mining which transforms extracted raw data into meaningful data. Raw data is mostly inconsistent,incomplete, may lack in certain behaviours and trends and may contain many errors. This process helps to resolve these type of issues. It basically prepares raw data for further processing and data transformation for efficient and effective processing for the purpose of the user.
In this section, we created an additional attribute named "Malicious" in the train dataset. For malicious sites the value of this attribute was set to 1 and for benign sites it was set to 0.This attribute was used as a target attribute for training the data. Similarly,we created a "Malicious" attribute for the test dataset using the same concept by placing 0 for benign sites and 1 for legitimate sites.However,this attribute was not part of the features used given as input for testing.It was used to check the accuracy of our model by checking this attribute against the predicted output of the test dataset by our algorithm.While calculating the site popularity features namely rank_country and rank_host of the url from www.alexa.com[2], for some urls it returned null values because those urls may not have any rank as they were not so popular.So, to deal with this situation we replaced the null value by -1.Generally, for popular urls we would get a definite value and for the urls which are not so popular like phishing sites we would get the value of this site popularity attributes to be -1. This would assist our algorithm to distinguish among malicious and phishing urls.
Further, we removed the features which were not in the suitable format for our algorithms like the non-numeric features. So, we removed the following attributes from our input features to algorithms as these were in string format:
1.host
2.path

These features were used to extract other features like rank_host,rank_country and after calculating this features we removed them to apply our algorithms.

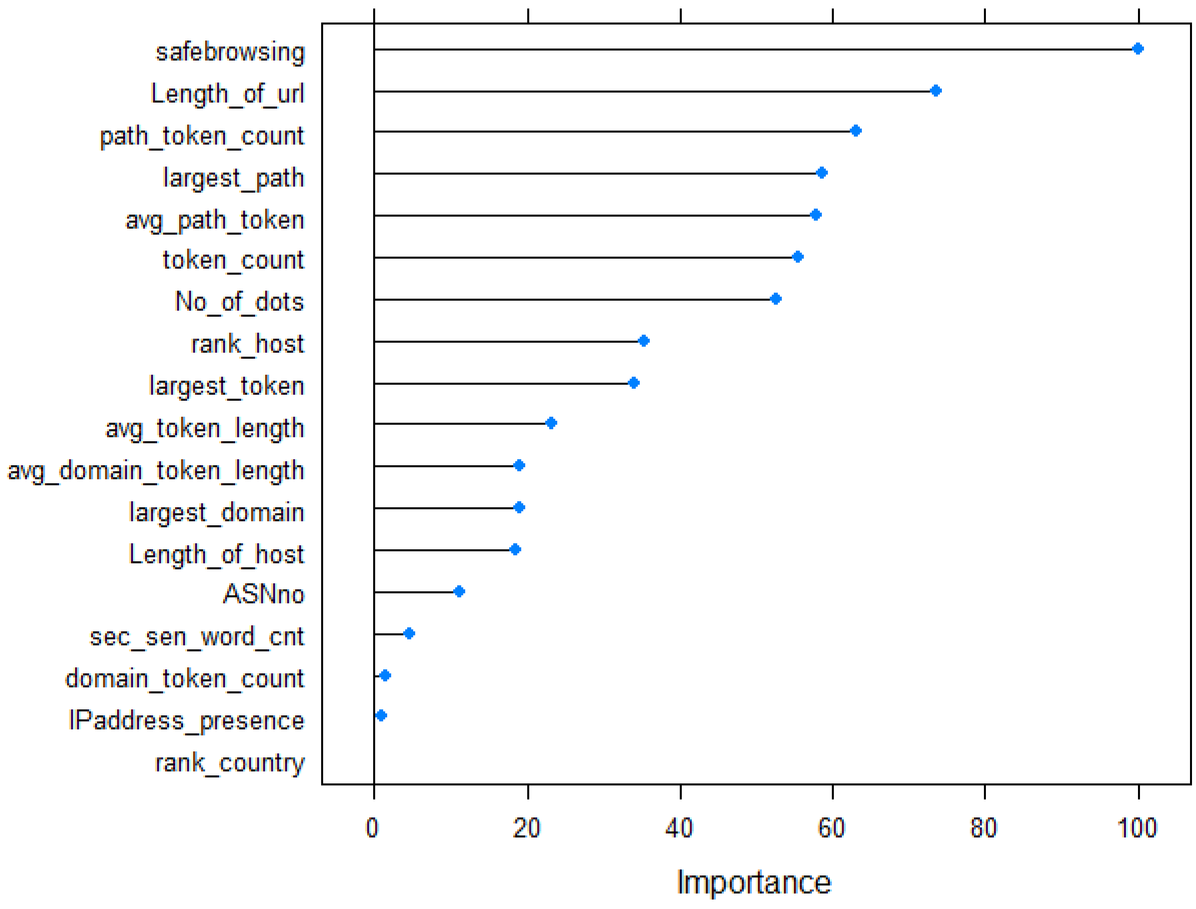


Figure 2.2: Feature Importance Plot

FEATURE_NAME	**Overall**
safebrowsing	100.0000
Length_of_url	73.6643
path_token_count	63.1155
largest_path	58.6178
avg_path_token	57.7561
token_count	55.4415
No_of_dots	52.5427
rank_host	35.1897
largest_token	34.1273
avg_token_length	23.1351
avg_domain_token_length	19.1165
largest_domain	18.9563
Length_of_host	18.4654
ASNno	11.0968
sec_sen_word_cnt	4.5085
domain_token_count	1.4666
IPaddress_presence	0.9333
rank_country	0.0000

Table 2.1: Feature Importance

After calculating the feature importance we can see that rank country is not an important feature to train so we removed rank country feature from our dataset.

2.1.1.5 Data Integration

In this Method Data is combined from different online sources into a coherent store . Integrating metadata from different sources under schema integration. The most impotant role of data integration is to remove redundancy and duplicacy in data.

2.2 Implementation

2.2.1 Planning and Requirement Analysis

It is the most important stage to develop the quality of our product.The software and hardware used in our project are as follows:
Software used:
The software used are as follows:
1.Ipython Notebook(python 2.7)
2.R studio
Hardware used:
No such specific hardware was required in our experimental methodology.

2.2.2 System Architecture

Figure 2.3 demonstrates the review of the framework architecture.This framework will use supervised offline machine learning algorithms; subsequently, it should be educated with a groups of classified information in order to build a picture of the classifier i.e. before utilizing the tool it should take some time in learning stage, in which it would find out about the two classes. In the learning stage, an accumulation of sites is given, at that point they are set apart as either genuine or phishing. The entire collection of vectors is sustained to ML engine and it will produce a classifier. There are two stages: learning and the testing(detection) stage. In the learning stage, the framework finds out about how to differentiate the phishing and genuine site. This learning is entirely based on the features that are extracted using URLs and these features indicate if a website is benign or phishing. ML motor uses the information gave in the vectors and by having the knowledge of phishing status, i.e. being phishing or genuine,of the website it adapts each class is having what properties.
Detection framework has one input and one outcome, it takes a URL to a site and chooses its phishing status and yields the prediction. Once the learning stage is done, the framework can be utilized. At the point when given the URL, the framework will extract the phishing features out of it, like one in learning stage, at that point utilizing the classifier produced in the past stage it will give the judgment for the asked URL.

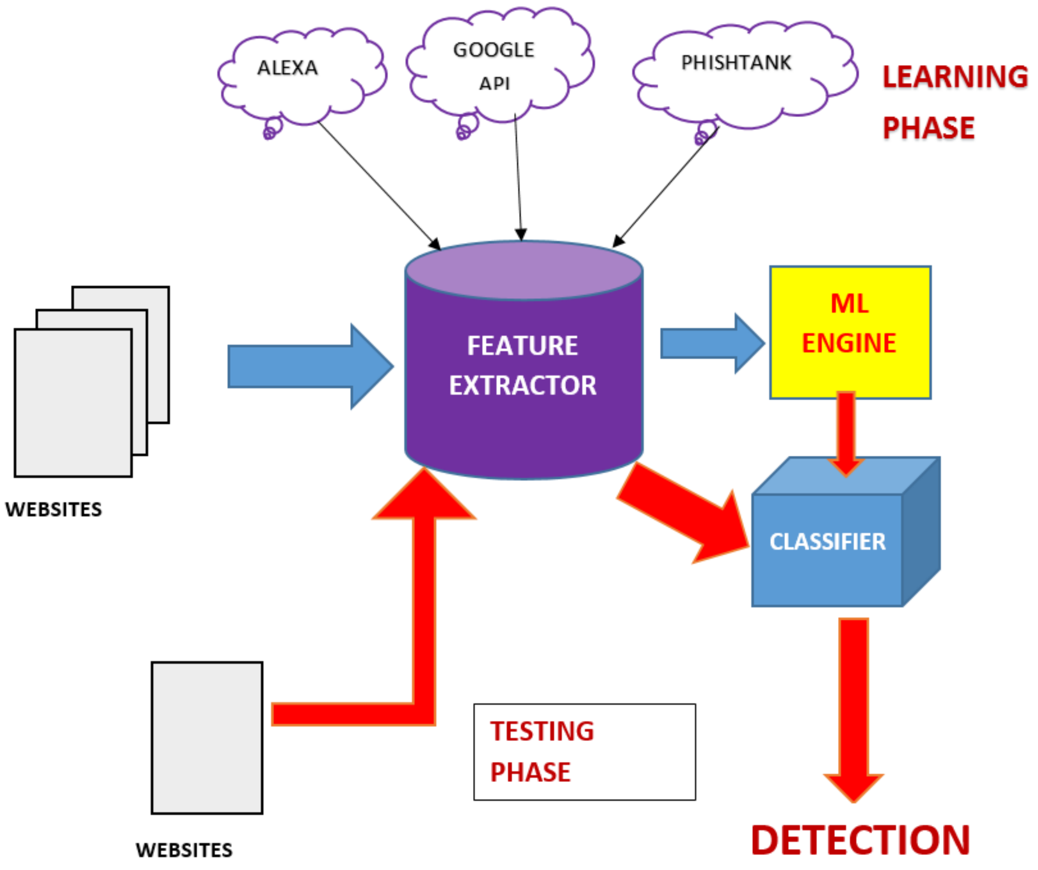


Figure 2.3: System Architecture

2.2.3 Methodology

We applied the following algorithms on our extracted data after cleaning and processing it:

2.2.3.1 Decision Tree Classifer

Decision tree is a classification based learning method that was presented by Quinlan (A. K. Shrivas, 2015). It makes a tree shape for classifying samples.Tree internal nodes relates to a feature,and the edges from the node splitt the information as according the value of the features (A. K. Shrivas, 2015). Decision tree consists a decision area and leaf node. In this method, we split the population or test into at least two homogeneous sets according to most critical splitter(i.e differentiator) in input variables(features). Data type is not a limitation in these models.
The decision area checks the condition of the samples and separates them into each leaf node or the next decision area. The decision tree is very fast and easy to implement; however, it has the risk of over fitting.

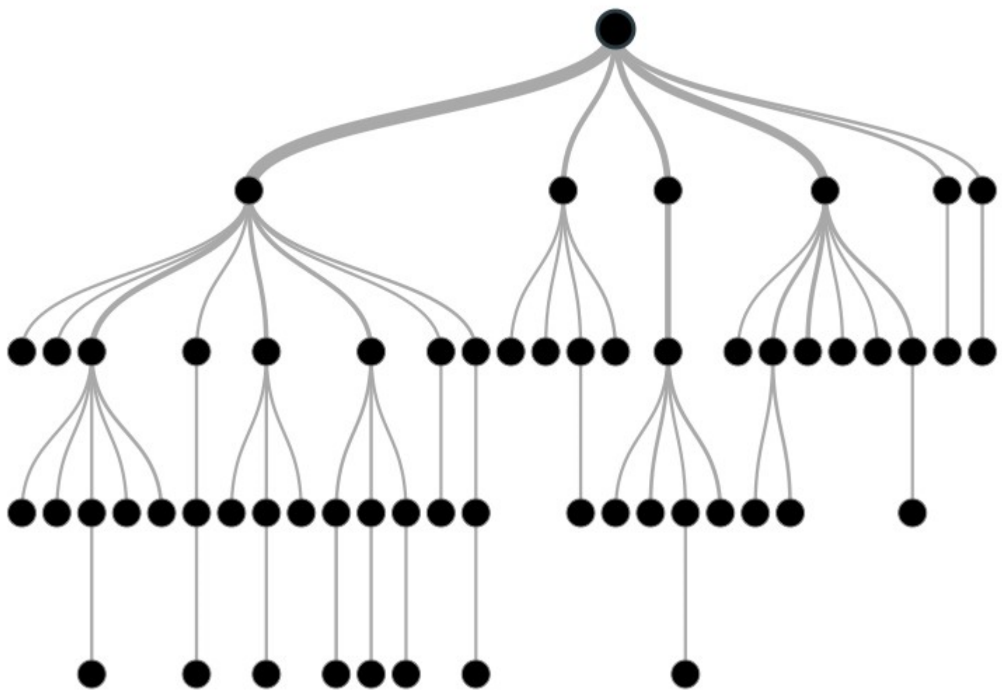


Figure 2.4: Decision tree Classifier

The libraries required were imported as follows:

from sklearn import preprocessing
from sklearn.tree import DecisionTreeClassifier
The model was applied as follows:
deci = DecisionTreeClassifier(random_state = 100,max_depth=3, min_samples_leaf=5)
deci.fit(train[train_cols], train['malicious'])
The predictions were made as follows:
query['result']=deci.predict(query[train_cols])

2.2.3.2 Treebagging Classifier

Tree Bagging classifier is a machine learning ensemble meta-algorithm. This system enhances the strength and exactness of machine learning algorithms utilized in classification problems and regression problems. It is a way to reduce the variance of the prediction. It generates the additional data for training from the original dataset using different combinations. It helps in overcoming the shortcomings of other machine learning algorithms too. It combines the individual predictions from the subsets (that are formed using original dataset) to form the final prediction for the dataset.This algorithm creates the new training datasets from the original training dataset. Some perceptions might be repeated in every subset. Let n be the size of original dataset and n' be the size of new training sets. If n is equal to n' then for large value of n the set each subset is expected to have (1-1/e) observations which are unique in it while the others are expected as duplicates. This type of sample that is formed is called Bootstrap Sample. Then these Bootstrap samples are fitted using different models.

The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn.ensemble import BaggingClassifier

The model was applied as follows:
bag = BaggingClassifier(n_estimators=150)
bag.fit(train[train_cols], train['malicious'])
The predictions were made as follows:
query['result']=bag.predict(query[train_cols])

2.2.3.3 Random Forest Classifier

Random forest is a supervised tree-based classification algorithm.It can be applied for both the regression and classification type of problems. This algorithm creates the forest with a number of trees.Generally,in the random forest classifier, more the number of trees in the forest it will give the high accuracy results.
this consists of many decision trees. all of them may use different approaches, but each of them select attributes randomly. The main advantage of this type of classifier is that they are having high precision and speed too but they can sometime suffer from over-fitting which can be solve by cross-validation method. Numbers of trees and the number of attributes used in each of the tree are the options of this classifier.

The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn.ensemble import RandomForestClassifier

The model was applied as follows:
rf = RandomForestClassifier(n_estimators=150)
rf.fit(train[train_cols], train['malicious'])
The predictions were made as follows:
query['result']=rf.predict(query[train_cols])

2.2.3.4 Artificial Neural Networks

ANNs are a kind of model which processes data just like the way a human mind does the processing of data. Neural Networks provide an effective solution for the problems of pattern recognition and classification. To train a neural network is a complicated task but it is very vital part of supervised learning. One of the most interesting feature that neural network has, is the possibility of learning. The weights associated with the interconnections among the elements and the characteristics of these elements regulate the outcome of transformation.These weights are used to store knowledge(*Statistics about phishing activity and phishtank usage*, 2013). The ANN analyses the information and produces a probability estimate that is matched with the data the algorithm has been trained to recognize. The algorithm learns by initially training the system by having both the input and output of the problem that is being required to solve. The network configuration is refined till acceptable outcomes are achieved. As the examples of known results are fed to the network, it gains experience.The algorithm manipulates the weighting factor so that to achieve the final output near to the result that one desire to achieve. As ANN possesses the ability to do multi-class classifications, a single

ANN (Scaled Conjugate Gradient), is used for classification, using the same training and testing data sets.

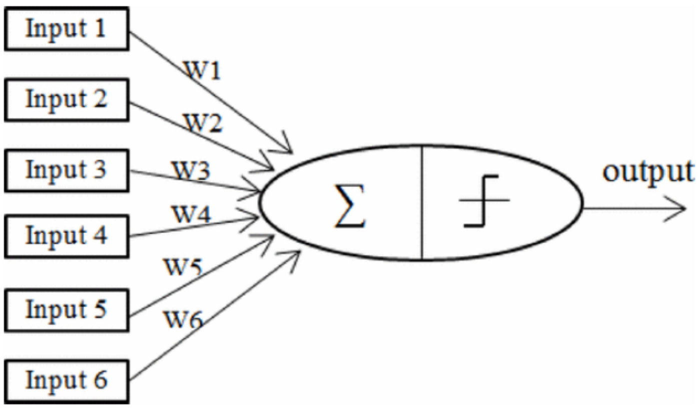


Figure 2.5: Neural Network

The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn.neural_network import MLPClassifier

The model was applied as follows:
clf = MLPClassifier(alpha=1e-5, hidden_layer_sizes=(40,20), random_state=1)
clf.fit(train, target)
The predictions were made as follows:
y=clf.predict(test)

2.2.3.5 Logistic Regression Classifier

We applied the logistic regression based algorithm with L1 regularization (Mitchell, 1997)(Schmidt, 2007).We have given brief overview of this algorithm and explain how its application can be used in the scam detection.Each dataset of websites is represented as D_k which is a set of feature vectors x_i.

$$D_k = \{ (x_i = (1, x_{i1},x_{i2},....x_{im}) \}_{i=1}^{n}$$

x_{ij} is the values of feature j of feature vector x_i.
Let $y_i \in$ {-1(not scam),1(scam)} are the labels for the website that is represented by x_i

.In logistic regression algorithm, the probability of website being scam is:

$$P(y_i = 1 \mid x_i , r) = \frac{1}{1+exp(r.x)}$$
$$\text{Where r.x} = \sum_j x_{ij}\, r_{ij}$$

The training stage procedure will learn by maximizing the following function with being the regularization parameter and $||r||_1 = \sum_j |r_j|$ is the L1-norm of the weight vector:

$$||r||_1 = \sum_j \log (P(y_i|x_i,r) - \lambda\, ||r||_1$$

This type of objective function has L1 regularization term $\lambda\ ||r||_1$ and the term which is negative in the maximization issue, keeps the weights from getting too extensive and stays away from over fitting. the procedure of optimization with L1 regularization is not the same as usual gradient ascent because of the fact that l(r) is not differentiable at focuses where weights are zero. We have utilized the projection approach by (Schmidt, 2007) because of its quick convergence and scalability. with the addition of preventing over fitting, L1 regularization produces sparse weight vector ; i.e, the subsequent vector contains a large number of zeroes. The benefit of the sparse weight solution is that the the features which are having zero weight are not very affection the classification decision, which implies that we can exclude those features in our data gathering. An attractive property of this logistic algorithm for our errand is that the classifier yields the likelihood of the site being scam (from Equation 1). To make the expectation about another site, we initially need to make feature vector x_{query} . r utilizing Host Analyzer, at that point we figure x_{query} . r and then contrast it with a pre-set limit:

$$f(x) = \begin{cases} \frac{x^2-x}{x}, & \text{if } x \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

We used the threshold of (t = 0) in experiments which corresponds to P (y_i=1| x_i, r) = 0.5. i.e, we will conclude that the website will be scam if it is more than 50% . We can adjust the threshold value to maintain the desired precision and recall trade-off.

The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
The model was applied as follows:
logis = LogisticRegression()
logis.fit(train[train_cols], train['malicious'])

The predictions were made as follows:
query['result']=logis.predict(query[train_cols])

2.2.3.6 K-Nearest Neighbors

K-Nearest Neighbors is a classification algorithm which has been successfully applied to various information-extracting problems(Huh and Kim, 2012).It performs the clustering of the elements having same charcterstics. It decides the class category of a test example taking its k neighbor that is near to it as the basis. It classifies the input data using k training data that is similar to the input data. KNN makes use of Euclidean distance to calculate the similarity between the input and training samples. The performance of this algorithm is mainly determined by the choice of k.The value of k in the KNN depends on two things: (a) size of dataset and (b) the classification problem type (Huh and Kim, 2012).
KNN algorithm works in the following manner:
Firstly, using Euclidean distance for calculating the distance, the nearest element from the test data a to training data K is to be found. For two elements in k dimensional space, a = [a1, a2, ... , ak] and y = [b1, b2, ... , bk], the Euclidean distance based on the two elements can be computed by using (1) :

$$\text{d(a,b)} = \text{sqrt}(\textstyle\sum_{i=1}^{k} (b_i - a_i)^2)$$

After the collection of k nearest neighbours is done, the k-nearest neighbors which are in majority will be considered as a class This considered class is used for the test data. Figure 2.6 illustrates the fact that KNN classifies the target on the basis of its neighbors.

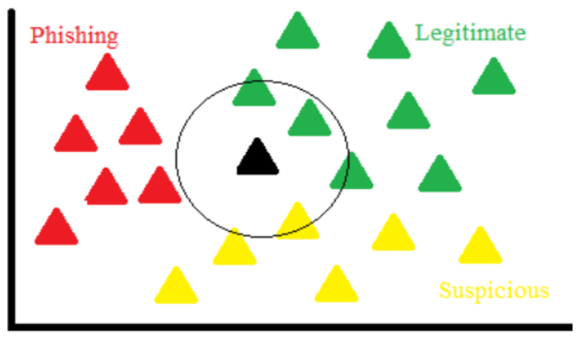


Figure 2.6: A K-Nearest Neighbor Classifer

The libraries required were imported as follows:
from sklearn import preprocessing

from sklearn.neighbors import KNeighborsClassifier
The model was applied as follows:
Kneigh = KNeighborsClassifier()
Kneigh.fit(train[train_cols], train['malicious'])
The predictions were made as follows:
query['result']=Kneigh.predict(query[train_cols])

2.2.3.7 SVM classifier

SVM (Support Vector Machine) which is a supervised machine learning mechanism which can be used for both classification or regression challenges. However, it is used in the classification problems most of the time. It is considered a statistical claasifier. All the Statistical classifiers are learned before they are used. The meaning is that a SVM classifier is fed with a number of classified elements; since classifiers assume that elements of each class is similar to each other's and different to ones of other classes. Therefore when a statistical classifier comes in to action, it does a comparison between the elements that it faces in learning phase and once which are given in the detecting phase and finally predicts. In SVM, every element is shown with a point in n dimensional space; where n=number of properties defined for each element.
In the first phase i.e. learning phase, a number of elements which are having known classification are provided to SVM. Svm tries to differentiate between the two classes of elements by drawing a line. The purpose of this line is to segregate the elements belonging to different classes from each other. For drawing the best line, it choses the one which consists of the most of the margins of both sides. All elements of each class are assumed to be similar so this line covers for the probable noise that exist over each element. Selecting the best set of properties is very important as the performance of this classifier depends vastly on it. Properties should be of different nature in different classes. It there is such a property which has similar value for two classes then property serves no use for Support Vector Machine.
The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn import svm
The model was applied as follows:
clf = svm.SVC()
clf.fit(train[train_cols], train['malicious'])
The predictions were made as follows:
query['result']=clf.predict(query[train_cols])

2.2.3.8 Gradient Boosting Classifier

Gradient boosting is an effective approach for building predictive models. The idea of boosting is based on if a poor learner can be trained to become better. We can say that "it is an efficient algorithm to convert relatively poor hypotheses into very good hypotheses".
It is a powerful machine learning algorithm to tackle problems like classification & regression, that gives a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Similar to the various boosting algorithm frames ,it also frames the model in a step-wise fashion , and it infers them by granting the enhancement of an arbitrary differentiable loss function.

Th algorithm consists of three elements:
1.Loss function which is to be optimized.
2.Weak learner to make predictions.
3. An additive model to add weak learners to optimize the loss function.

1.An advantage of the this framework is that any new boosting algorithm need not be evaluated for every loss function that need to be used, instead, it is a generic framework that any differentiable loss function can be used.
2.As the weak learner Decision trees are used in gradient boosting. In each step, it introduces a weak learner to compensate the weaknesses of existing weak learners.
3.Trees are added one at a time, and existing trees in the model are not changed.
When adding trees a gradient descent procedure is used to minimize the loss .Here, gradient descent is used to minimize a set of parameters, like weights in a neural network or the coefficients in a regression equation . After calculating loss or error, the weights are updated to minimize that error.

Gradient Boosting = Gradient Descent + Boosting

In Gradient Boosting,“weaknesses” are identified by gradients.
The libraries required were imported as follows:
from sklearn import preprocessing
from sklearn.ensemble import GradientBoostingClassifier
The model was applied as follows:
grad = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0)
grad.fit(train[train_cols], train['malicious'])

The predictions were made as follows:
query['result']=grad.predict(query[train_cols])

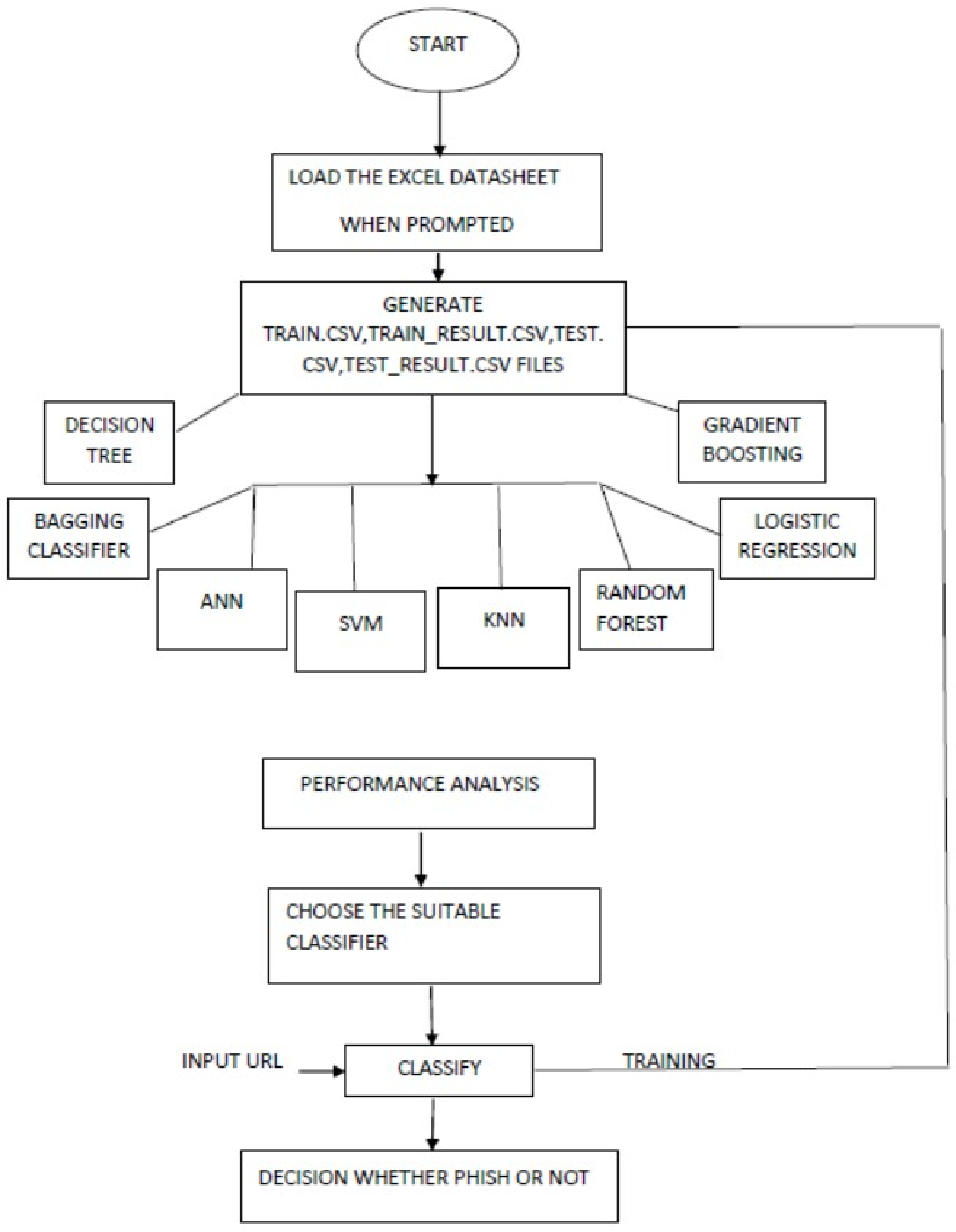

START
LOAD THE EXCEL DATASHEET WHEN PROMPTED
GENERATE TRAIN.CSV,TRAIN_RESULT.CSV,TEST. CSV,TEST_RESULT.CSV FILES
DECISION TREE
GRADIENT BOOSTING
BAGGING CLASSIFIER
ANN
SVM
KNN
RANDOM FOREST
LOGISTIC REGRESSION
PERFORMANCE ANALYSIS
CHOOSE THE SUITABLE CLASSIFIER
INPUT URL
CLASSIFY
TRAINING
DECISION WHETHER PHISH OR NOT

Figure 2.7: Design Details and Implementation Flowchart

2.3 Graphic User Interface Templates

We have prepared GUI in python(Tkinter library) using which the user can verify if the url suspicious to him/her is legitimate or malicious.It is user-friendly and effective way to prevent the internet users from phishing attacks.

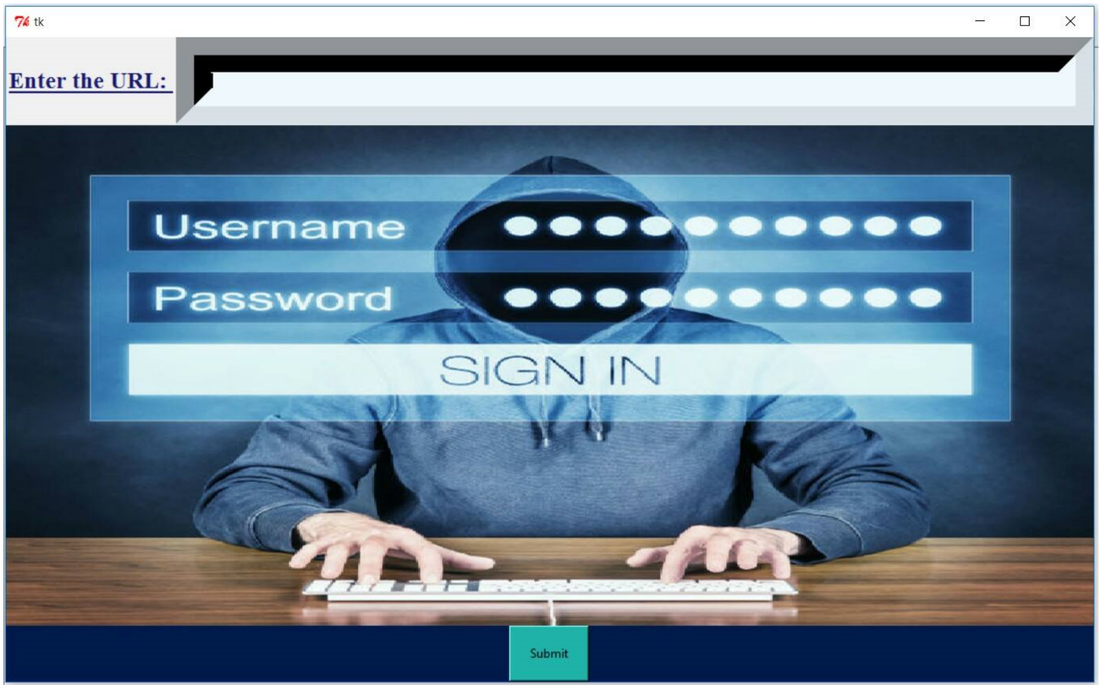
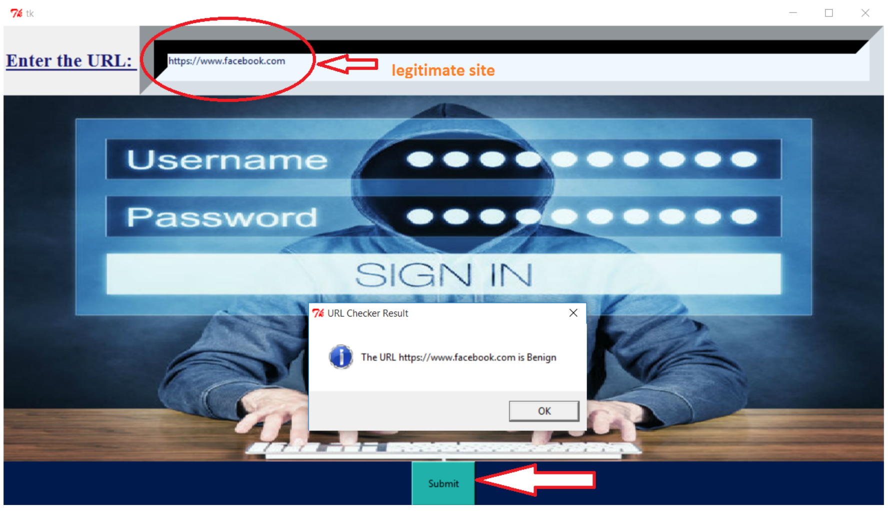


Figure 2.8: Graphical User Interface Template

Our GUI has a text box in which user can enter the url for which he/she has to check the phishing status. On clicking the submit button on the gui ..it will give the result as:

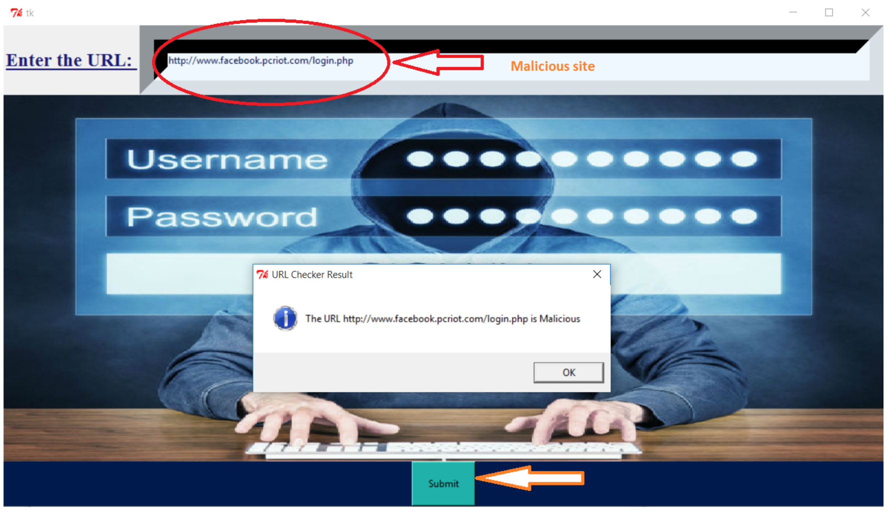
1.The URL (www.something.com) is Maliocious(if it is phishing website)
2.The URL (www.something.com) is Benign(if it is genuine website)



The URL https://www.facebook.com is Benign

Figure 2.9: GUI Implementation of Benign URL

In figure 2.9:
The url (http://www.facebook.com) is written in the text box .As this is the genuine website as we know so the result shown here is :
"The URL (http://www.facebook.com) is Benign"



The URL http://www.facebook.pcriot.com/login.php is Malicious

Figure 2.10: GUI Implementation of Malicious URL

In figure 2.10:
The url (http://www.facebook.pcriot.com/login.php) is written in the text box .As this is the phishing website as we know so the result shown here is :
"The URL (http://www.facebook.pcriot.com/login.php) is Malicious"-

CHAPTER 3

RESULTS AND DISCUSSION

3.1 RESULTS

Responses to user requests are delivered to the users using Graphical interface. User needs to enter the URL in the text box of GUI and then user will get to know the phishing status of the URL he entered.

PERFORMANCE METRICS:
We used these metrics to calculate the accuracies of our applied models.

True Positive (TP) - Number of phishing website recognized correctly
False Positive (FP) - Number of legitimate website recognized incorrectly as phishing website
True Negative (TN) - Number of legitimate website recognized correctly
False Negative (FN) - Number of phishing website recognized incorrectly as legitimate website

Table 3.1: Performance metrics

After training our datasets with various supervised algorithm and after getting the prediction we got the following accuracies:-

Confusion matrix of decision tree classifier

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
Prediction Outcome	**NO**	True negative=77	False Negative=23	100
	YES	False Positive=373	True Positive=477	850
	total	450	500	

Table 3.2: Confusion matrix of decision tree classifier

The accuracy obtained after applying Decision Tree Classifier was 58.32%

Confusion matrix of Tree Bagging Classifier

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
Prediction Outcome	**NO**	True Negative=92	False Negative=11	103
	YES	False Positive=358	True Positive=489	847
	total	450	500	

Table 3.3: Confusion matrix of Tree Bagging Classifier

The accuracy obtained after applying Tree Bagging Classifier was 61.16%

Confusion matrix of Random Forest

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
	NO	True Negative=142	False Negative=12	154
Prediction Outcome	**YES**	False Positive=308	True Positive=488	796
	total	450	500	

Table 3.4: Confusion matrix of Random Forest

The accuracy obtained after applying Random Forest Classifier was 66.32%

Confusion matrix of Neural Network

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
	NO	True Negative=437	False Negative=94	531
Prediction Outcome	**YES**	False Positive=13	True Positive=406	419
	total	450	500	

Table 3.5: Confusion matrix of Neural Network

The accuracy obtained after applying Artificial Neural Network was 88.74%

Confusion matrix of Logistic Regression Classifier

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
	NO	True Negative=442	False Negative=81	523
Prediction Outcome	**YES**	False Positive=8	True Positive=419	427
	total	450	500	

Table 3.6: Confusion matrix of Logistic Regression Classifier

The accuracy obtained after applying Logistic Regression Classifier was 90.63%

Confusion matrix of KNN

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
	NO	True Negative=416	False Negative=46	462
Prediction Outcome	**YES**	False Positive=34	True Positive=454	488
	total	450	500	

Table 3.7: Confusion matrix of KNN

The accuracy obtained after applying K-Nearest Neighbors Classifier was 91.58%

Confusion matrix of Support Vector Machine

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
Prediction Outcome	**NO**	True Negative=435	False Negative=51	486
	YES	False Positive=15	True Positive=449	464
	total	450	500	

Table 3.8: Confusion matrix of SVM.

The accuracy obtained after applying Support vector machine was 93.05%

Confusion matrix of Gradient Boosting Classifier

		Actual outcome		
	N=950	**NO**	**YES**	**Total**
Prediction Outcome	**NO**	True Negative=423	False Negative=18	450
	YES	False Positive=27	True Positive=482	500
	total	450	500	

Table 3.9: Confusion matrix of Gradient Boosting Classifier.

The accuracy obtained after applying Gradient Boosting Classifier was 95.26%

After checking all the accuracies of different models , finally we decided gradient boosting classifier is best in all of them which has a accuracy of 95.26%

3.2 DISCUSSIONS

The increase in online industries has resulted in an increase in number of phishing assaults throughout the years.
As indicated by the measurements, one of 257.9 emails leads to phishing sites in oct 2012. The majority of them focused on the financial, payments and retail benefits. In beginning of 2012, web clients has lost around 686 million dollars in the phishing assaults .Therefore it turns out to be most critical to build up a quick and exact phishing recognition tool. *Statistics about phishing activity and phishtank usage* (2013)
The GUI of our phishing detection system engages end users and provides them an environment for detecting malicious sites
The whole discussion is to provide a user-friendly and effective, efficient way to prevent the internet users from phishing attacks and protect them from malicious sites.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

Insufficient knowledge and consciousness on phishing education makes such malicious attacks successful. Even the few indicators used by the browser such as pad lock identification, lock icon, and site identify button, don't help and the user still cannot identify the attack. Additionally, the users should not follow links to sites blindly where they need to enter the delicate information.It is vital to check the URL before entering the site.
The principal motivation of this study was to assist the users in analysing the legitimate web page and fake web page by using URL as an indicator. We proposed a url based phishing detection system using lexical features, site popularity features and host-based features. Our test results demonstrated that our proposed approach was very successful in preventing phishing assaults, as 95.26% phishing sites were identified precisely by utilizing the proposed system. Moreover,our methodology uses the Uniform Resource Locator (URL) itself without accessing of Web sites and analyzed it. In this manner, it wiped out the runtime idleness and the likelihood of exposing users to the program based vulnerabilities and hence provided a effective technique to shield the web users from phishing.

4.2 Future Scope

For future works, our approach can be learned and tried against real world datasets; as the dataset utilized by us comprises of just predetermined number of sites, which may create biased results. subsequently, our approach ought to be tuned for all websites, including some of new low profile genuine sites which are falsely recognized as phishing

in our approach.
The approach we proposed was focused mainly on a GUI level phishing detection system. But the broad picture can be seen as follows:

- Instead of gui , we can build an extension which can work online.This will be more user friendly and more suitable for the real time environment.

- In the future, the predictions and detection accuracies of the proposed approach can be further improved by taking the other features along with the lexical,host-based and site-popularity features .In any case extracting different other features will influence the running time complexity of the system and would increase it.

- Also, in the future we can add additional constraints in the program and can check more source codes that contains several languages in it like PHP, JAVA, CSS,ASP, Perl, asp etc.

- Moving forward we can incorporate different parts of web based learning and assembling information to see the new patterns in phishing exercises like the quick changing DNS servers.

REFERENCES

[1] A. Fu, L. W. and Deng, X.: 2006, Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd), *Dependable and Secure Computing, IEEE Transactions* **3**, 301–311.

[2] A. K. Shrivas, R. S.: 2015, Decision tree classifier for classification of phishing website with info gain feature selection, *IJRASET* .

[3] A.Bavani, D.Aarthi2, V. C.: 2017, Detecting phishing websites on real time using anti-phishing framework, *Department of Information Technology (UG) 1, 2, 3, Assistant Professor4 Kingston Engineering College, India* .

[4] Adulghani Ali Ahmed, N. A. A.: 2016, Real time detection of phishing websites, *IEEE* .

[5] Afroz, S. and Greenstadt, R.: 2015, Detecting phishing websites by looking at them, *IEEE Communications Society* .

[6] Anjali B. Sayamber, A. M. D.: 2014, On url classification, *International Journal of Computer Trends and Technology (IJCTT)* **12**.

[7] Chowdhury, N. K. and Islam, M.: 2006, Phishing websites detection using machine learning based classification techniques, *Department of Computer Science Engineering, University of Chittagong, Bangladesh* .

[8] *Google safe browsing API*: 2015.

[9] Hongl, J.: 2009, A hybrid phish detection approach by identity discovery and keywords retrievall, pp. 571–580.

[10] Huh, J. and Kim, H.: 2012, Phishing detection with popular search engines : Simple and effective, *FPS'11 Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security* **8**, 194– 207.

[11] Jain, A. and Gupta, B.: 2016, Eurasip journal on information security, *EURASIP Journal* pp. 426–436.

[12] Longfei WC, J. W.: 2016, A lightweight anti-phishing scheme for mobile phones, *(IEEE: 2016)-MobiFish:* .

[13] M. Khonji, A. J. and Iraqi, Y.: 2011, A novel phishing classification based on url features, *GCC Conference and Exhibition (GCC)* **6**, 221–224.

[14] Mitchell: 1997, *T. M. Machine learning*, McGraw-Hill,New York.

[15] Mona Ghotaish Alkhozae, O. A. B.: 2015, Phishing websites detection based on phishing characteristics in the webpage source code, *Department of Computer Sciences,FCIT King Abdulaziz University, Jeddah, KSA* .

[16] *Phishtank: www.phishtank.com*: 2017.

[17] Schmidt, M., F. G. R.: 2007, Fast optimization methods for l1 regularization: a comparative study and two new approaches.

[18] *Statistics about phishing activity and phishtank usage*: 2013, *Proc. of IEEE Wireless Communications and Networking Conference* .
URL: *http://www.phishtank.com/stats/2013/01/*

[19] *The Web information company –www.alexa.com*: 2017.

[20] W Liu, X Deng, G. H. A. F.: 2006, An antiphishing strategy based on visual similarity assessment, *IEEE Internet Comput* .

[21] Yue, C. and Wang, H.: 2010, Bogusbiter: A transparent protection against phishing attacksl, *ACM Transactions on Internet Technology* **10**.

[22] Zhang, J.: 2010, An content-analysis based large scale anti-phishing gateway.

[23] Zhang, J. and Wang, Y.: 2012, A real-time automatic detection of phishing urls.