CHAPTER 1

INTRODUCTION

Web assumes an imperative part in the today's reality which motivates unauthorized access to the information, internet is growing rapidly day by day which motivates the number of malware software's. Malicious is a virus in which intentionally interrupting the user (or) accessing the personal information as an unauthorized user. Malware is a software which disrupts (or) damages to a computer system. Any one wants to search a file (or) any website they search with the help of URL name , In the address space bar give the URL name it will take you to the particular page (or) website .

The standard syntax for the URL: <Protocol><Hostname><path>

Example http://mail.google.com/mail/#inbox, Http specifies the standard protocol, Mail.google.com specifies the host name, and Mail/#inbox specifies the path name. Malicious websites are the main backbone and cornerstone for all the internet criminal activities, the danger of these sites may damage the user's information we need to protect the end users from visiting this sites. Pernicious sites cover a scope of various endeavor which are dangerous to visit that is the reason distinctive sorts of noxious destinations apportions different dangers to clients.

Three noteworthy classes of malignant locales

- spamming (sending unwanted email messages to the users)
- Phishing (Identified with the false email in which honest client gets caught by pernicious sites)

- Malware (It is to disrupt (or) damage a computer system).A guileless SVM and Random Forest is an order system which is basically in view of the SVM hypothesis with suspicion of freedom among indicators. It is a conditional probability not dependent on each other.

Malicious websites supports a variety of socially undesirable properties such as spam advertised commerce and malware propagation the most common problem is that user visit there sites. There visits can be results by messages web query items (or) site page links but the user has to take some actions, user must decide regardless of whether to tap on a new URL if the client click in the noxious URLs it will take user to the dangerous websites.

Mindful of this troubles, numerous security analysts had created different framework to shield the client from these malevolent sites. The user they do not have any information regarding the malicious URLs it seems to look like real URLs but when the user clicks on that it will damage the system so the user should have awareness about this.

URLs are the intelligible content strings which are parsed standardly by the customer programs. Let us see how the URL works the browsers translates each URL in to the instruction which locates the server hosting and determine where the site is put on that host. We have the accompanying standard sentence structure.
<Protocol>://<hostname><path>

The protocol specifies which standard convention ought to be utilized to recover the asked for asset The most commonly used protocol is hypertext transfer protocol (HTTP).

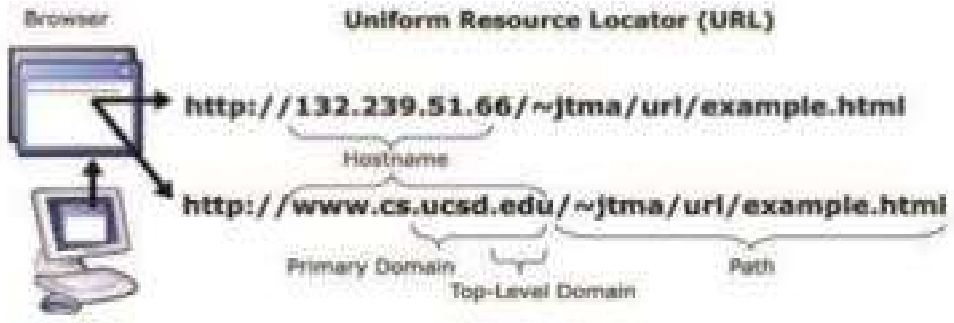


Fig.1.1 URL Example

In the fig 1 the way is /jtma/url/example.html the tokens are delimited by slashes, dots and da shes., dots and dashes. Which indicates how the site is deals with.

The <hostname> which is used as a identifier for the web server at times it is called as IP address, IP address is a network protocol which is used to send the system parcels of one host to the another host. The key idea behind the IP address is all host on the web contains the IP address which enables them to reach one another.

1.1 Why the Svm and Random forest has chosen?

Support vector machine (SVM):

SVM is a grouping calculation utilized for the practical edge that finds hyper plane among two classes which distinguish positive and negative.

We plot every information thing with the estimation of each component being the estimation of a specific co-ordinate.

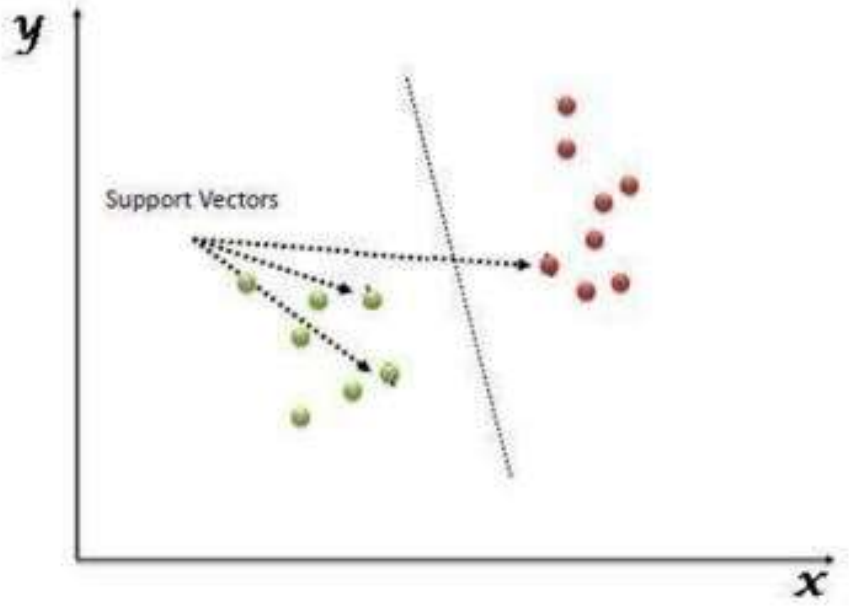


Fig1.2 SVM Segregates two classes

1.2 How does the Svm works.?

From the above we know how to isolating hyper plane with the help of two classes. By and by the question is distinguish the privilege hyper plane. For this we have four scenarios described below. We need to identify the right hyper planes for that we have like four scenarios in that describes how each scenario chooses the right hyper planes.

Identify the right hyper plane(scenario-1)

In this case we are having a three hyper planes A, B and C, we have to recognize the privilege hyper plane to arrange the elements. Choose the hyper plane which separates the two classes in a better way. In this hyper plane B has selected.

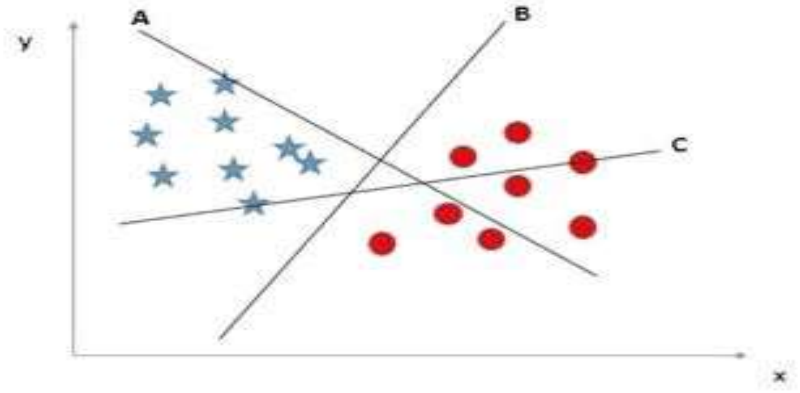


Fig 1.3 Identify right hyper plane

Identify the right hyper plane(scenario-2)

Here we have three hyper planes A, B and C and every one of the classes are differentiating well.

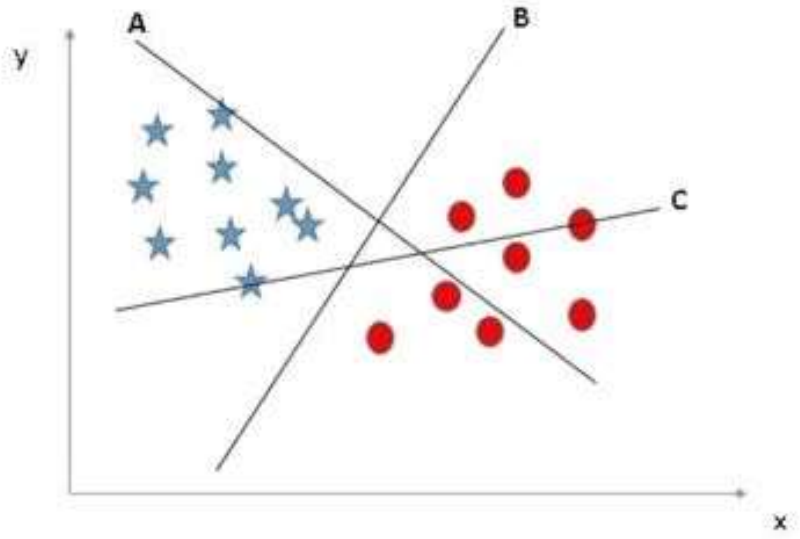


Figure 1.4 Identify the right hyper plane(scenario-2)

Enhancing the dividing between nearest data point and the hyper plane will help us to select the benefit hyper plane. The detachment is called as edge, the edge for hyper plane C is high when compared to both A and B. So select hyper plane C.

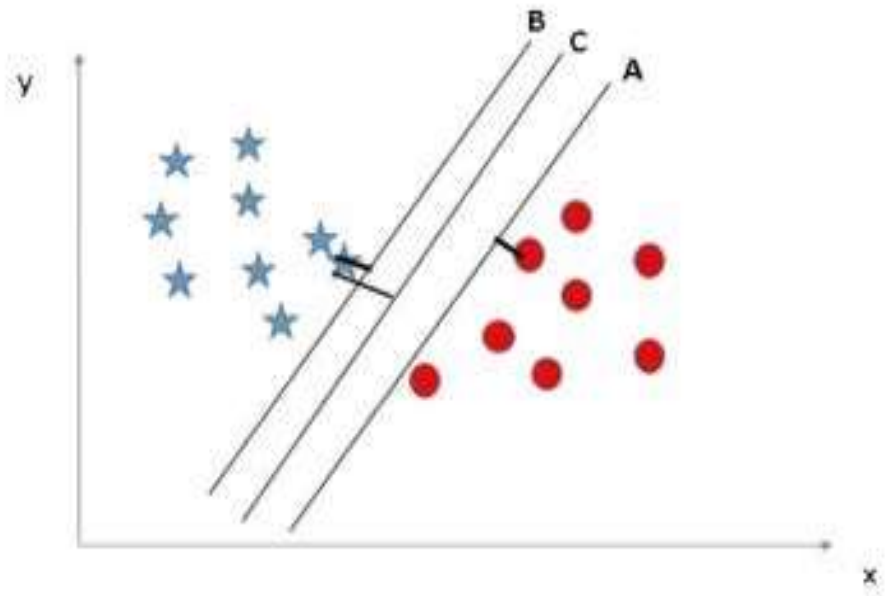


Figure 1.5 Identify the right hyper plane (scenario-3)

Around there you may have selected the hyper plane B since it has higher edge contrasted with B however the SVM chooses which orders the classes precisely to boosting edge.

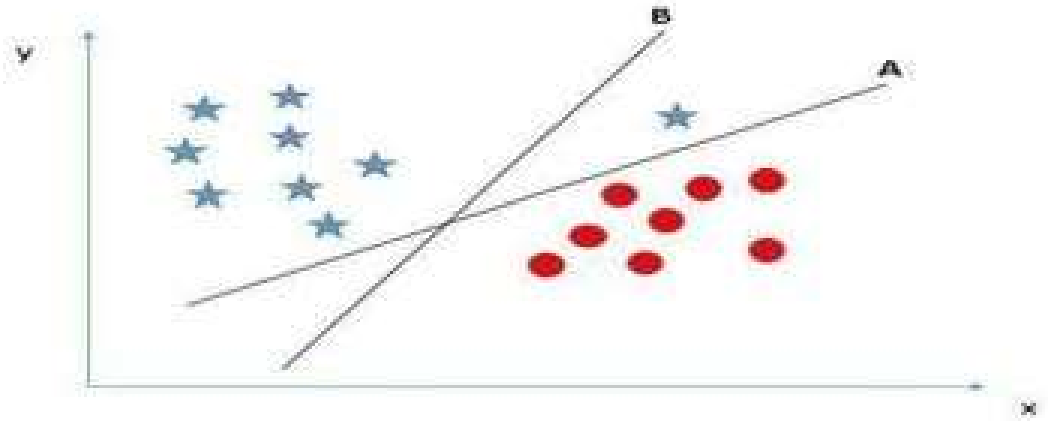


Figure 1.6 Identify the right hyper plane (scenario-4)

Hence forth the hyper plane B has mistaken and A which has characterized every one of the classes effectively so the privilege hyper plane is A.

Identify the right hyper plane (scenario-4)

From the figure we can say that we are not able to isolate the two classes utilizing a straight line.

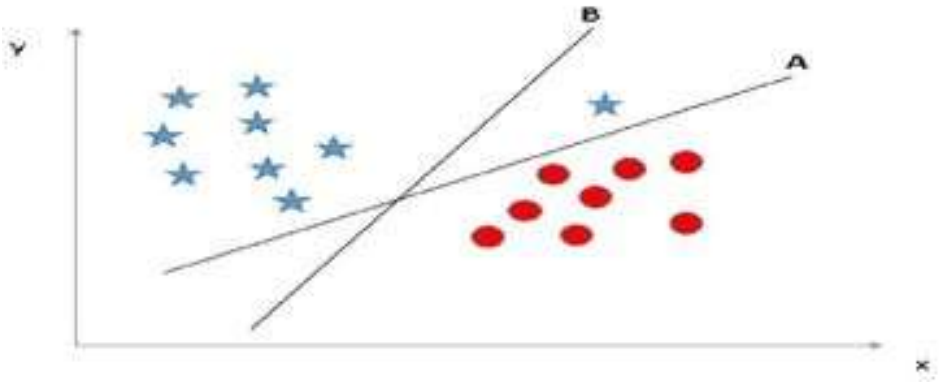


Figure 1.7 Identify the right hyper plane (scenario-5)

In this section one star at flip side resembles an exception for star class, SVM has a component to disregard anomalies.

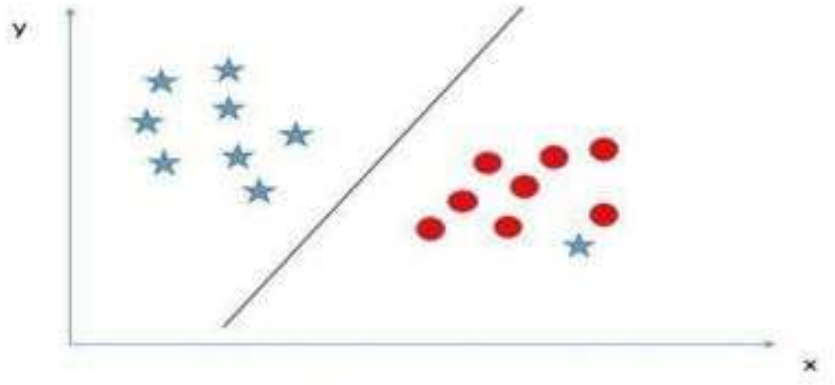


Figure 1.8 Identify the right hyper plane

1.3 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at trainin g time and outPutting the class that is the mode of the classes (classification) or mean predicti on (regression) of the individual trees. Random decision forests correct for decision trees' habi t of overfitting to their set. The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implem ent the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, and "Rando m Forests"is their trademark. The extension combines Breiman's "bagging" idea and random s election of features, introduced first by Ho and later independently by Amit and Geman in ord er to construct a collection of decision trees with controlled variance.

CHAPTER 2

LITERATURE SURVEY

This area gives a diagram of the research that has been done in the area of malicious links and which forms the basis for the implementation of this project. The main source of knowledge and inspiration for this project comes from. This section provides an overview of this publication.

Numerous specialists have taken a shot at the discovery of suspicious URLs, we introduce a more comprehensive arrangement of elements, we are preparing a bigger datasets. Kan and Thi [2005] he made a review on the machine learning for URL order. To prepare the model rapidly they dissect the lexical components of URL strings. To generate the features what they did they just utilize a large number of features in to a tokens, and length of each part of the URL.

When comparing the lexical components and the page content elements they get the accuracy of 95%. The technique called as logistic regression for classifying the phishing URLs, Thus they have included the feature called as red flag keyword in URL features such as page rank, web page quantity guidelines they had achieve a 97.3% accuracy over 2500 URLs.

McGrath and Gupta they does not use any classifier instead they had compared the examination of phishing and Non phishing URLs with the assistance of datasets, they had taken the Non phishing URLs shape the DMOZ open registry extend and phishing URLs are taken from Phish tank and a Non open source.

2.1 Problem Definition

This project presents different angles related with the URL (Uniform Resource Locator) portrayal handle which sees whether the target site is a malicious or good. The standard datasets are used from different sources. The rising issue spamming, phishing and malware, has made a necessity for strong framework course of action which can describe and also recognize the bad URL. After classifying the malwares the result should be displayed according to the classification of the malware. This project proposes to implement a SVM and Random forest classifier. Upon implementation, at a very high level, there will be following components that will be available

1. First the dataset is collected from the different sources

2. Next train the SVM and Random forest module with the help of datasets

3. Extract the features of the training dataset and store it in the csv file

4. After extracting the features testing the data and again train those test data and give it to the SVM and Random forest module.

5. The SVM and Random forest module says whether the target website is benign or malicious.

6. If the target website is benign then it will pop up a message as the particular URL is benign that means the URL is free from malicious.

7. If the target website is said to be malicious then it has to classify three types of malicious data whether it is Phishing or malware.

The issue indicates the URL arrangement can be spoken to as a parallel issue where the positive cases are malignant. The technique can be effective if there are the qualities for vindictive and generous URLs are not quite the same as each other. To generate the features what they did they just utilize a sack of words depiction of the tokens in the URL.

Mainly there are three steps they are

- First avoid page content downloading it is safe for users.
- Second classifying the URLs with the help of trained model and it is light.
- Weight operation when compared to downloading the page content.
- Lastly focus on the URLs features classifier which makes appropriate to any setting in which URLs are found.

Finally acquiring the malignant elements of page for both preparing and testing information.
Phishing emails are said to be a spam message. This emails usually used in the criminal fields that mainly relies on forged mail which is originating from a legitimate company (or) bank.

The phisher emails attempts the user to redirect to the malicious websites, that are designed to acquire the individual data, for example, usernames, passwords, Visa points of interest.
Phishing emails present a serious threat to the electronic commerce because this type of emails are used in individual and financial organization on the internet. According to the research made by the Gartner on this phishing attacks almost 3.6 million users in the US had lost their money to phishing attacks and the overall total losses had reached around to 3.6 million in 2007.

The federal trade commission receives a complaint from all the users and they recognize the phishing attacks (or) emails grouped first. Malicious emails can take the personal information from the users and that leads to the loss of funds.

This survey gives classified guide to present state of the evaluation and comparison of different methods on phishing emails detection are given important.

2.3 System Analysis of Existing System

In existing framework bolster vector machine is utilized for discovery and recognizable proof of kind of malignant URLs. The Naïve Bayes which is used for which it is utilized for the paired order issues.

Naïve Bayes is nonlinearly separable which means consider two classes which is not correctly classified then it is said to be a nonlinearly separable. The objective of a Naïve Bayes is to locate the ideal isolating hyper plane which boosts the edge of the preparation information.

The main thing we can see from this definition, is that a Naïve Bayes needs preparing information. Which implies it is a directed learning calculation. It is likewise critical to realize that Naïve Bayes is a characterization calculation. Which implies we will utilize it to anticipate if something has a place with a specific class. In existing system the training data will take more time to extract the features and duplicates of the data will be there in the datasets.

The datasets contains thousands of data each and every line it has to extract the features such has lexical features, web features, network features, link popularity features, DNS features, while extracting the features the bulk of data's is present in the dataset it needs to extract each and every data so it takes more time for processing this is the problem in existing system.

The SVM and Random Forest classifier has the capability to extract bulk of features and it tak es less time to extract the bulk of data features and duplicate data's is also avoided by using Random forest classifier so the Random forest and SVM classifier gives more accuracy when compared to the Naïve Bayes , This is the problem in existing system.

Disadvantages of existing system

- The training datasets contains thousands of data each and every line it has to extract the features such has lexical features, web features, network features, link popularity features, DNS features , while extracting the features the bulk of data's is present in the dataset it needs to extract each and every data so it takes more time for processing.

- Because the training dataset contains bulk data's and it contains huge amount of data there may be a chances of having the duplicate values.

2.4 Proposed System

In the proposed system the SVM and Random Forest Classifier is used for detecting and classifying the malicious URLs. It takes less time for extracting the features when compared to the Naïve Bayes, bulk of data's can be extracted with the help of Random Forest Classifier, even the duplicates will be avoided by using naïve Random Forest Classifier.

Credulous Random Forest is a Classifier which is used as a conditional probability in which features of one class is not dependent on one another means it is independent from each other. It gives more accuracy when compared to the Naïve Bayes.

Advantages of Aroposed System:

- The proposed system which uses a SVM and Random Forest Classifier for detecting and identifying the bad URLs. The proposed demonstrate in light SVM and Random Forest Classifier is upheld by grouping and arrangement method. The proposed system in this project shows that, we can use a large datasets can be used, SVM and Random Forest models got the hang of utilizing Probability display has preferable exactness over Naïve Bayes show. SVM and Random Forest classifier takes less processing time to extract the bulk features of data.
- The duplicates of the data's in the dataset also been avoided by using naïve Bayes classifier so we can say that SVM and Random Forest Classifier gives better accuracy when compared to the Naïve Bayes.

CHAPTER 3

REQUIREMENT SPECIFICATION

This requirement specification section provides the detailed description about the software that are required to implement the proposed prediction model, in order to do that we need some software components that performs the required operation and are compatible to the requirements. It is a way of recommending the tools that reduces the effort of developing the project.

3.1 Hardware Requirements

- **Hardware Configuration**

o	Processor	-	Pentium–IV
o	Speed	-	1.1 Ghz
o	RAM	-	256MB(min)
o	Hard Disk	-	20 GB

3.2 Software Requirement

- **Software Requirements**

o	Operating System	-	windows7/8/10 /Ubuntu 14.0
o	Programming Language	-	Python
o	Tools	-	Python (IDE)

3.3 Functional and Non-functional Requirements

The functional requirements are usually used to describe the requirements of the customer, like what all functionalities should the software should serve. Their expectation from the software. From the customer view, the software should be user friendly, it should not include complicated operations that user has to do manually, they expect it to be as simple as possible.

Functional Requirements of Proposed System

The proposed system accepts the input from the user and represents various graphs representing probabilistic results.

- **Input** – urls
- **Operation** – feed the extracted information to classifier/predictor

- **Output** –Results of comparing the features.

Non-Functional Requirements

Usability

The device is user-friendly to use, the user has to just load the dataset/ input the url name so that it gives the goal site is malicious or good

Availability

System will be available at any time and can be used in order to take some counter measures against attacks.

Portability

As the implementation is done using python IDE, the system is portable to other operating systems, but the required packages has to be imported.

Integrity

The model is developed using python, in order to make the system work well, all the required libraries has to be imported. The implementation of model is simple and understandable.

Extensibility

Since the project implemented is simple and understandable, it can be used to extend the functionalities and for future work.

3.4 Libraries from python

1. Pandas

Pandas is a python package that is designed for analyzing data It is a power library because of its flexibility and suggestibility.

- Pandas suits for following type of data
    - Data in excel sheet or SQL table
    - Unordered data
    - Ordered data
    - data
    - Matrix data

Data structures of pandas library are

- Data frames which is of 2Dimensional
- Series which is of 1Dimensional

Advantages of Pandas are

- Capability of handling missing values, these missing values will be represented as NaN that stands for Not a Number.
- Size of data frames can be changeable like new column can be inserted or deleted.
- Alignment of data.
- Slicing of data frames.
- Join and merge operations on dataset.
- Fast.

2. Numpy

A python package designed for performing scientific computations. Apart from scientific computations numpy has several functionalities.

- N dimensional array.
- Broadcasting.
- Linear algebra and Fourier transform.
- Integrating different programming languages.
- An array class of numpy is ndarray
- ndarray.dim – displays the dimension of an array.

3. Tkinter

- User to create simple user interfaces

- Used to create a dialog box, message box or to print the output in a window.

4. tkFileDialog

- In the event that you need to open or spare a document or to pick an index utilizing a file dialog you don't need to execute it all alone.

- The module tkFileDialog is only for you. Much of the time the seven comfort capacities given by the module will serve your necessities.

5. tkMessageBox

- The tkMessageBox module is utilized to show message confines your applications.

- This module gives various capacities that you can use to show a fitting message

- Some of these capacities are showinfo, showwarning, showerror, askquestion, askokcan cel, askyesno, and askretryignore

Parameters

Function Name: This is the name of the fitting message box work

title: This is the content to be shown in the title bar of a message box.

message: This is the content to be shown as a message.

Choices: choices are elective decisions that you may use to tailor a standard message box. A portion of the alternatives that you can utilize are default and parent.

- The default alternative is utilized to indicate the default catch, for example, ABORT, RETRY, or IGNORE in the message box. The parent choice is utilized to determine the window on top of which the message box is to be shown.

Accuracy score

- one of the performance metrics calculated by confusion matrix
- accuracy for multi-label/ multiclass classification
- parameters - actual values, predicted values

CHAPTER 4

SYSTEM DEVELOPMENT STRATEGY

4.1 Architecture of the Project

The architecture of this project is an adapted version of the architecture proposed. Following figure shows the various components being developed in this project

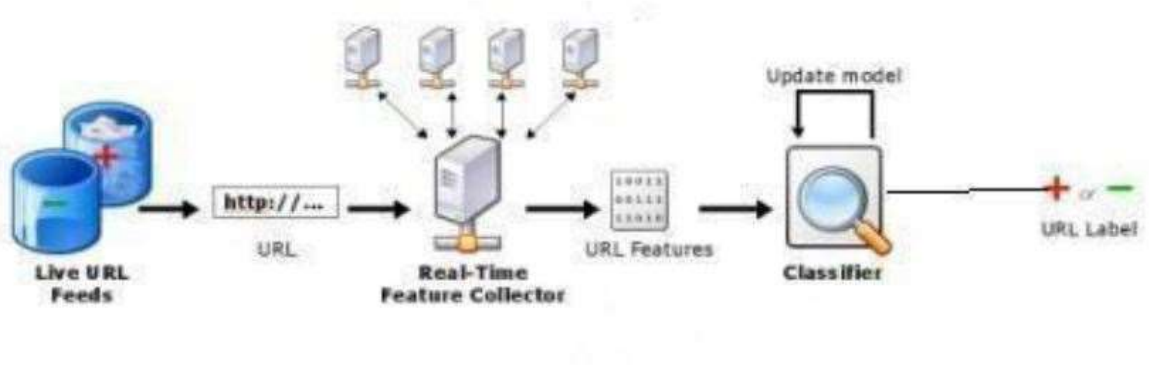


Fig 4.1: Proposed System Architecture

The units in Fig. 8 which describes the architecture of the proposed system

1. Live URL feeds: This is the first component in the architecture diagram where all the different data's are collected from the different sources. The dataset contains two attributes URLs and labels and the datasets will be stored in the .txt file.

2. Training data: This is the second component in the architecture diagram where all the data which is present in the dataset should be given to the module for training purpose. The module should be trained. The main stage in framework configuration is preparing information which comprises of taking after strides.

A general practice is to part your information into a preparation and test set. You prepare/tune your model with your preparing set and test how well it sums up to information. Your model's execution on your test set will give bits of knowledge on how your model is performing and enable you to work out issues like predisposition versus fluctuation exchange offs.
Like all analyses, more often than not you will need to do arbitrary examining to acquire preparing and test sets that are pretty much illustrative populace tests. However you should be aware of issues like class imbalance when your number of observations in your dataset is very small, there have also been strong cases made to not split the data as less data will have impact on the predictive power of your model.

Training data is the data on which the machine learning programs learn to perform correlational tasks (classify, cluster, learn the attributes, et al). Testing data is the data, whose outcome is already known (even the outcome of training data is known) and is used to determine the accuracy of the machine learning algorithm, based on the training data (how effectively the learning happened).Basically you have three data sets: training, validation and testing. You train the classifier using 'training set', tune the parameters using 'validation set' and then test the performance of your classifier on unseen 'test set'. An important point to note is that during training the classifier only the training and/or validation set is available. The test set must not be used during training the classifier.

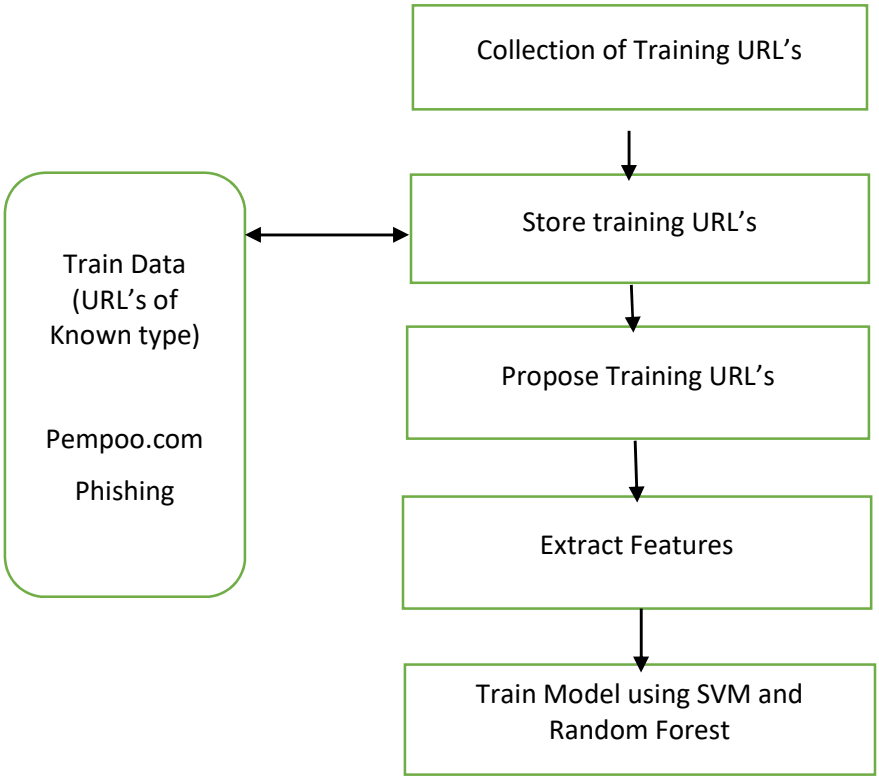


Fig 4.2:Training Data

First train the data using SVM and Random forest classifier and all the training data's will be collected and store the training URLs. Case: Pempoo.com phishing in this e.g. Pempo.com which indicates the domain name whereas the phishing which indicates the class which is of known URL. All the URLs are put together as a prepare set according to their classes.

After storing the training URLs give it to the extract features, in extract features all the features are extracted there are mainly five features namely lexical features, web features, network features, DNS features, link popularity features all these features are extracted and train the classification model it classifies whether the target website is benign or malicious.

If the target website is said to be malicious then it has to classify two types whether it is phishing or malware. Training the classifier is the first step in the design system.. There have also been strong cases made to not split the data as less data will have impact on the predictive power of your model.

3. Real time feature extractor: This is the third component in the architecture diagram which is used for extracting the features. In this project mainly there are six features we are extracting such as Lexical Feature, Link Popularity Features, Webpage Content Features, DNS Features, and Network Features.

- **Lexical features:** Lexical elements are just the properties of the URL which is not referencing the page. These lexical properties include some of the features such as host name length, the whole URL length, and number of dabs, number of slices, question marks, number of commas, these are genuine esteemed components.

  We make a twofold components for each and every tokens in the host name which is delimited by dot and the URL is delimited with the help of the features. This is known as sack of words. We don't save any request of the tokens, we simply make a qualification between tokens having a place with the host name, way URL, and the space names. In lexical features we have considered some of the important features such as average domain token length, domain token count, largest domain, average path token, path token count, largest path.

- **Web features:** web based elements depict where the malignant locales are facilitated on the site, and their identity overseen by and how they are administrated. We utilize these elements in light of the fact that the majority of the vindictive sites are facilitated in a less legitimate facilitating administrations.

The web pages are more exploited to the attackers to inject malicious code in to the w ebsites in order to avoid this we used some alternatives the clients tend to check the qu antities of HTML labels, iframe, lines, and hyperlinks inside the site page content. In web we have considered some of the important features such as the features are taking from the internet and consider the features iframe count, line count, hyper link count.

- **DNS features:** Domain name servers are the unit associated with the names, if the user forgets the IP address not to worry with the help of domain name server they can search , even remembering the IP address is difficult for the user so instead of that they can use the domain name servers and easily they can access the information of that particular site is difficult to remember the IP address of the particular website so instead of IP address we can use DNS, It is easy to remember the DNS names for example Google IP address is difficult to remember instead of that we can give names this is called DNS. From this we can say that most of the malevolent locales is being sent from an IP address space. The main important feature in this section is IP address.

- **Link popularity features:** This feature defines amount of incoming links from the alternative website, in this case we are considering the links which is having the highest popularity when compared to other links for example take google.com now a days all the people are using the Google as a search engine so we can say that Google is more popularity than any other links so we can give highest rank to those which is having popularity.

  Malignant connections have a less measure of connection ubiquity though the amiable connections have a tendency to have a most elevated connection notoriety. Here we considered the rank host as an important feature based on the rank we come to know which is having the highest link popularity and other features such as phishing join proportion, spamming join proportion and malware interface proportion.

- **Network features:** Attackers hide their malicious websites with the help of redirection counts, the malicious sites redirection count is different is from the benign site, both malicious and benign sites have different redirection counts, we can identify with the help of redirection count which is malicious site and which is benign site. Here we consider the components, for example, redirection check, space query time, download speed, download bundle content length, genuine downloaded bytes are the vital elements.

  We examine lexical and have based components since it contains more data about the URL the below figure shows a high level illustration of how an individual feature vector is constructed. The justification for using lexical features is that

  URLs to malicious sites tend to “look different” in the eyes of the users perspectives.

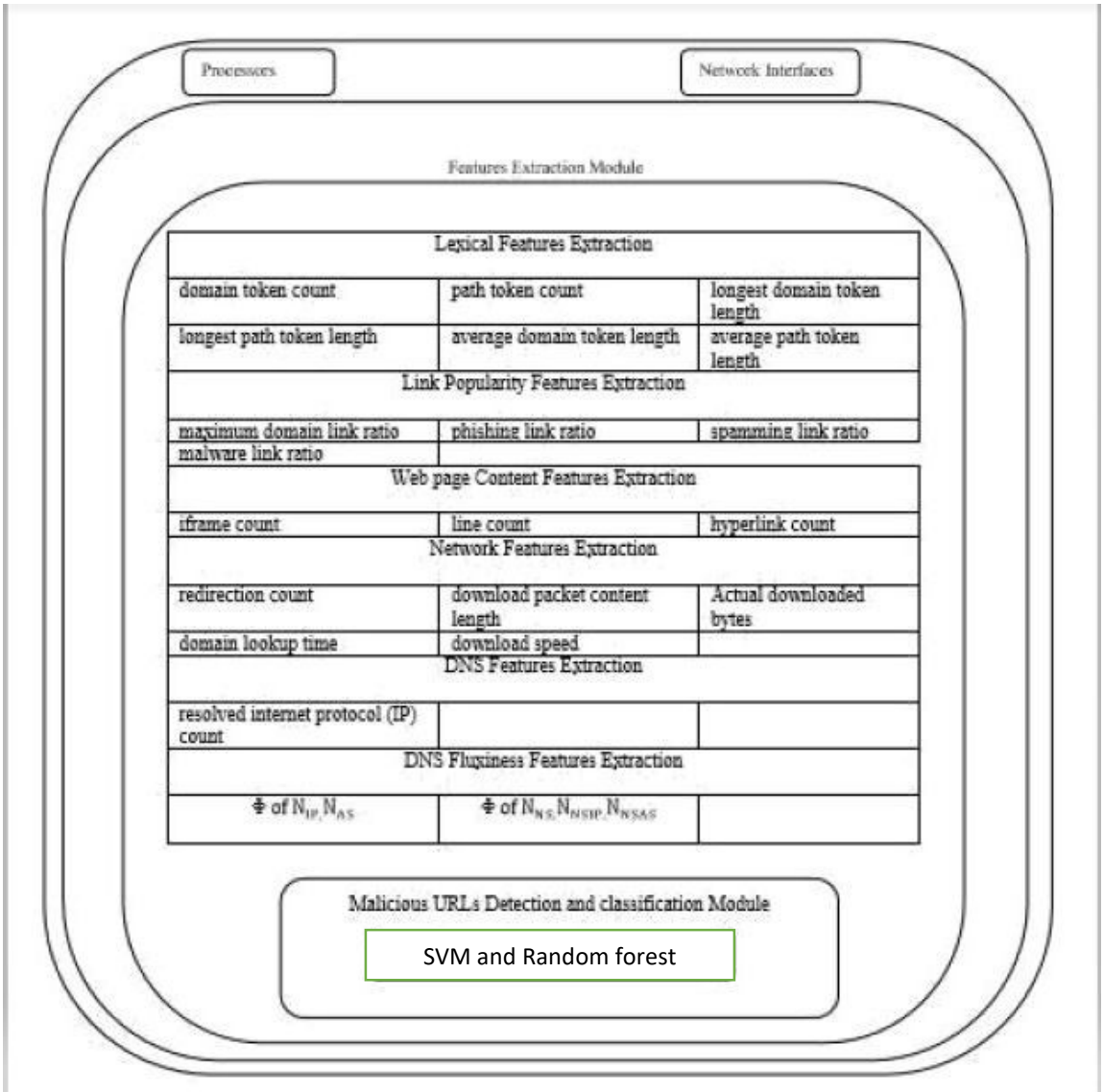


Fig 4.3:Feature Extraction Module

We develop the component vector for every URL continuously, when the element gathering gets a URL it endeavors to question a few servers to develop the host based information the host based which gathers all the information which has the segments. If we need to identify the IP address we just scan the IP prefix table with the given IP address.

For IP address highlights we scan for the IP prefix and connected with the given IP address, we gather the DNS information in light of the fact that the DNS is organized to deal with hig h volume. It requires to store the URL includes in a Euclidean space different components vec tors are put away as the lines and segments of a meager framework.

4. URL features: This is the fourth component in the architecture diagram, after extracting th e features test those URLs and give it to the classifier.

5. Classifier: This is the last component in the architecture diagram which is used for the g rouping whether the objective site is malevolent or not. In the design system comprises of the accompanying strides. In the figure the unknown URL is given whose domain name and class is given and need to identify the type of URL, unknown URL is given for the testing and sep arating the components related with URL and guide the elements with extricated highlights fro m obscure prepare set.

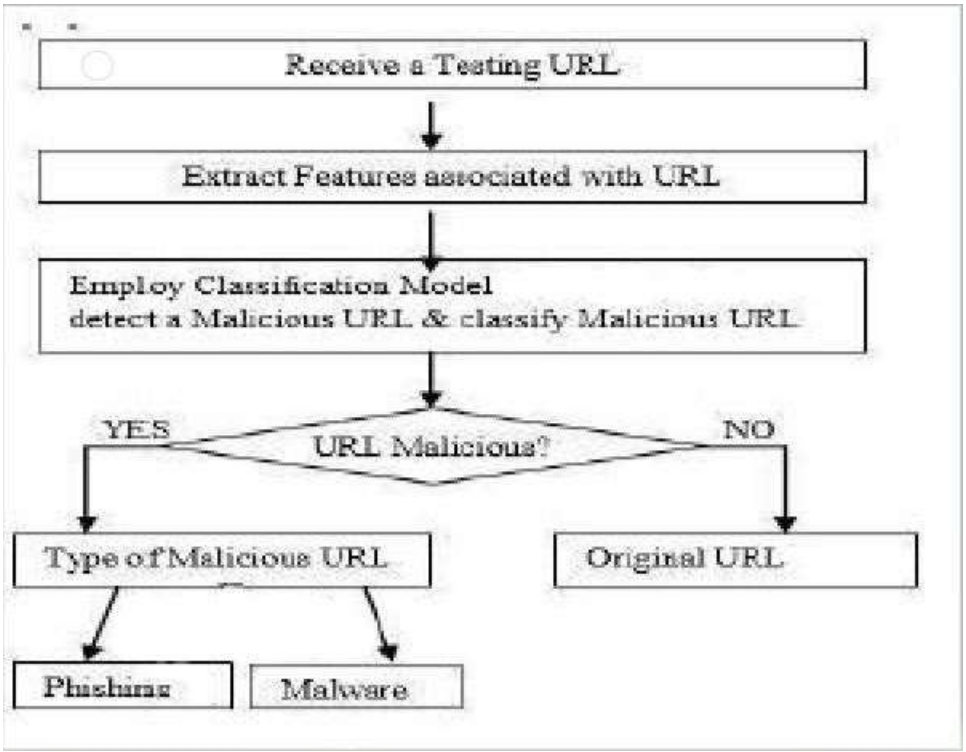


Fig 4.4:Classification Module

Receiving URL: URL is taken from the particular dataset or program.

Choosing four components of URL: In this section the elements are selected from the URL which is present in the dataset, for example, Domain, Primary Domain, Sub Domain and Path Domain. From that point onward, they are isolated into various segments.

Calculating six estimations of the heuristics: In this section, we select a whitelist which contains the Primary Domain of the genuine site as appeared. On the off chance that if the URL is negative, then the site is said to be a malicious site.

On the off chance that each esteem is sure,the site is considered as the honest to goodness site .

HTML Tags Features: Sometimes just the URL components are insufficient anticipate the si te is phishing or genuine. Once the webpage is named suspicious phish by Classification, mor e powerful or HTML label components are required to be separated from the source code of t he website page. For this component extraction we will utilize the HTML DOM-

Tree Parser which will empower us to see the HTML code and concentrate different subtle ele ments effortlessly.

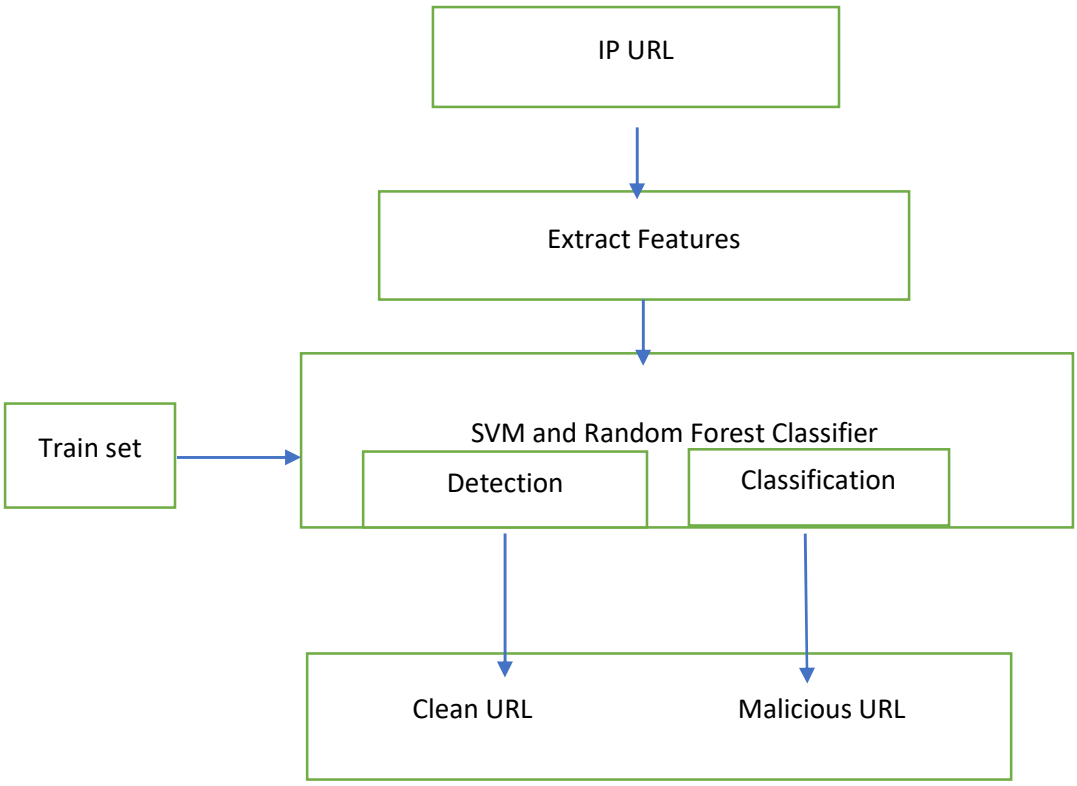
4.2 Domain's Ranking Features

Furthermore, clearly the phished sites are neither gotten to by the clients nor connected by alternate sites. Along these lines, the recurrence that the sites are gotten to by clients or connected by different sites can be used to choose whether the site is phishing or not.

Page Rank: PageRank which is used by the Google web search engine. The page rank for most of the phishing sites is low in light of the fact that these locales exist just for a brief span. Consequently, PageRank is used to know whether the page is phishing page or not. From this we can say that the phishing pages have the low page rank whereas the genuine pages have the highest page rank.

Alexa Rank: Alexa which indicates the rank of site in view of traffic data, get to levels, connections to different sites and the refreshed data. Most phishing sites exist just for a brief span so its Alexa Rank esteem is low because of our try different things with dataset from Phish Tank. We used Alexa Rank to know whether the site is phishing page or not. Same as the PageRank, Alexa Rank also produces low rank for phishing sites and high if there should be an occurrence of honest to goodness sites. At the end of the day, Alexa Rank esteem likewise assumes a key part in identifying phishing locales.

4.3 The Proposed Method Framework



4.5 Proposed method framework

The technique of proposed structure comprises of taking after techniques:

I/P URL: Collecting all the URLs from different data sources and give it to the next method called as extract features.

Extract Features:

- **Train set:** After extracting the features train the naïve Bayes classifier with the help of dataset which we are having.

SVM and Random Forest Classifier: Preparing the svm and random forest module and the classifier will detect and classifies the URLs, if the URL is clean then it is called as benign URL and if the URL is malicious then it has to classify the two types whether it is phishing or malware.

At the situation when svm and random forest model is used for selecting from the existing dat aset, it does not contains a greater number of qualities than segments accessible and it ensure that suspicion will be raised successfully. The proposed work in this paper speaks to trial asse ssment for arrangement and discovery of URLs utilizing svm and random forest.

4.4 Data Collection

A general practice is to part your information into a preparation and test set. You prepare/tune your model with you're preparing set and test how well it sums up to information. Your model's execution on your test set will give bits of knowledge on how your model is performing and enable you to work out issues like predisposition versus fluctuation exchange offs.

The real-
time gathering is important since we need to get the elements of URLs when they were first c

aught by the sustain. For every approaching URL, our element authority quickly questions DN S, WHOIS, boycott, and geographic information servers, and furthermore dealing with IP addr ess-related and lexical-
related components. After storing the training URLs give it to the extract features, in extract f eatures all the features are extracted there are mainly five features namely lexical features, we b features, network features, DNS features, link popularity features all these features are extra cted and train the classification model it classifies whether the target website is benign or mali cious.

4.5 Feature Collection Infrastructure

We construct the unit structure for each URL logically. Right when our component gathering server gets a URL, it attempts to scrutinize a couple of external servers to build up the host-based section of the segment vector. The host-based component amassing has the going with parts.

For the IP address we are going to search the IP prefix table and AS which is related with the given IP address, this can be downloaded from the Route Views Project. Before the accumulation starts the database has to be in the order stored in the memory.

The search for IP prefixes and AS numbers once the database is stacked in memory,. Regardless, specialists should remain up with the most recent by irregularly downloading the latest RIBs from Route Views. For WHOIS data, we developed a line PHP script around the PHP.

The script which manages the parsing of the WHOIS sections, these are secured with the flat substance files and there is no standard arrangement. The WHOIS which searches the idleness operations which takes like 1-
3 seconds for the territory names. We set some time limit such as 7 seconds to avoid the gath ering of malicious information.

4.6 Feature Representation

In feature representation we define the classification calculations in mat lab, which allows the user to store the features of URL in a high dimensional Euclidean space as a sparse vectors since new segments are continually created by effectively disguised URLs, the amount of areas in this cross section furthermore creates after some time.

In our use, we assemble data in well-ordered bumps the sparse vector of day N which contains the components of all the new elements which are gathered on the day N.

For cases that go before the first event of an as of late included component, we delegate zero regards to all components that have not yet appeared in the sustain. New components in like manner create new segments of the weight vector for direct classification. We also name zero regards to present the segments of weight vectors identifying with as of late included components.

For IP address highlights we scan for the IP prefix and connected with the given IP address, we gather the DNS information by parsing the yield if the host summons, the dormancy for th ese elements is normally low in light of the fact that the DNS is organized to deal with high v olume.

URL Features Used

Phishing URLs can be inspected in light of two sorts of elements: lexical elements and host-based elements of the URL. The lexical elements break down the configuration of the URL w hile the host based elements recognize the area, proprietor and how pernicious destinations are facilitated and overseen.

Lexical Features

It examinations the setup of the URL not the content of the page it references the URL. These properties incorporate the length of the whole URL, nearness of IP address in URL, the quant

ity of specks in the URL, nearness of phishing watchwords in URL, nearness of suspicious ch aracters, for example, @ image, hexadecimal characters and utilization of delimiters or extraor dinary double characters like "/", "?", ".", "=", "-
", "$", "^" either in the host name or way. It ought to be noticed that F1 to F8 are the elemen ts considered in this work.

- **Length of URL:** Most phishing URLs utilize extensive space names to bait end users so that the URL may seem honest to goodness. e.g.http://www.tsv1899benningenringen.de/chronik/refresh/alarm/ibclogon.php. Subsequently, if the length of a URL is longer than 55 characters, the URL is hailed suspicious.

- Utilization of IP address in URL: Some phishing sites contain an IP address in their URL rather than the area name with a specific end goal to shroud the real space name which is malevolent. At the point when the URL in an email has its host name as an IP address. For instance, in http://65.222.204.76/co/, we signal the URL suspicious.

- Utilizing the hexadecimal character codes: A malicious URL can likewise be spoken to utilizing hexadecimal base esteems with a "%" image to conceal the genuine letters and numbers in the URL. In this manner, a URL that has hexadecimal character codes will be hailed suspicious.

- Utilization of @ image in URL: The "@" character is utilized by phishers to make have names hard to get it. A @ image in a URL will empower the string to one side of the "@" image which is the real authentic URL to be disposed of while the string to the correct which prompts the phishing webpage is dealt with as the genuine site. For instance, in the URL http://www.worldbank.com@phishingsite.com, "www.worldbank.com" will be

disposed of while "phishingsite.com" will be dealt with as the real space name. At the point when a URL contains the '@'symbol is distinguished, we signal it suspicious.

- **Number of Sensitive Words in URL:** Some delicate words every now and again show up in phishing URLs, for example, secure, account, refresh, login, sign-in, saving money, affirm, confirm suspend, username, and so forth. URLs that contain at least one of these watchwords are regarded suspicious. We utilize this element to hail a URL as phishing.

Host Based Features

Host-based components depict the area of malevolent locales, that is, the place they are being facilitated, who these destinations are overseen by and how they are overseen. Some of these elements are time of area, page rank, number of spaces.

- **Age of Domain:** The age of the area distinguishes when a site is facilitated with the end goal that a site that has less age or is moderately new is hailed suspicious. Numerous phishing destinations have enlisted space names that exist just for a brief timeframe to sidestep identification. They might be as of late enrolled and a few areas may not be accessible at the season of checking. The WHOIS queries on the WHOIS server is utilized to recover the area enrollment date, and if the space enlistment section is not found on the WHOIS server, the URL is viewed as suspicious.

- **Presence of Form Tag:** One of the strategies phishers use to gather data from clients is the utilization of shape tag in URL. Hence, a URL that has the frame tag is hailed suspicious.

- **Number of Domains:** A phishing URL may contain at least two area names which are utilized to forward deliver from one space to the next. The quantity of area names in the URL separated from an email is numbered and if more than one, we hail the URL suspicious.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 INTERFACE

The following UI diagram shows the user can test the URLs by giving individual URLs.

I had given http://google.com where you can submit this URL and this you can see in the bel ow figure.

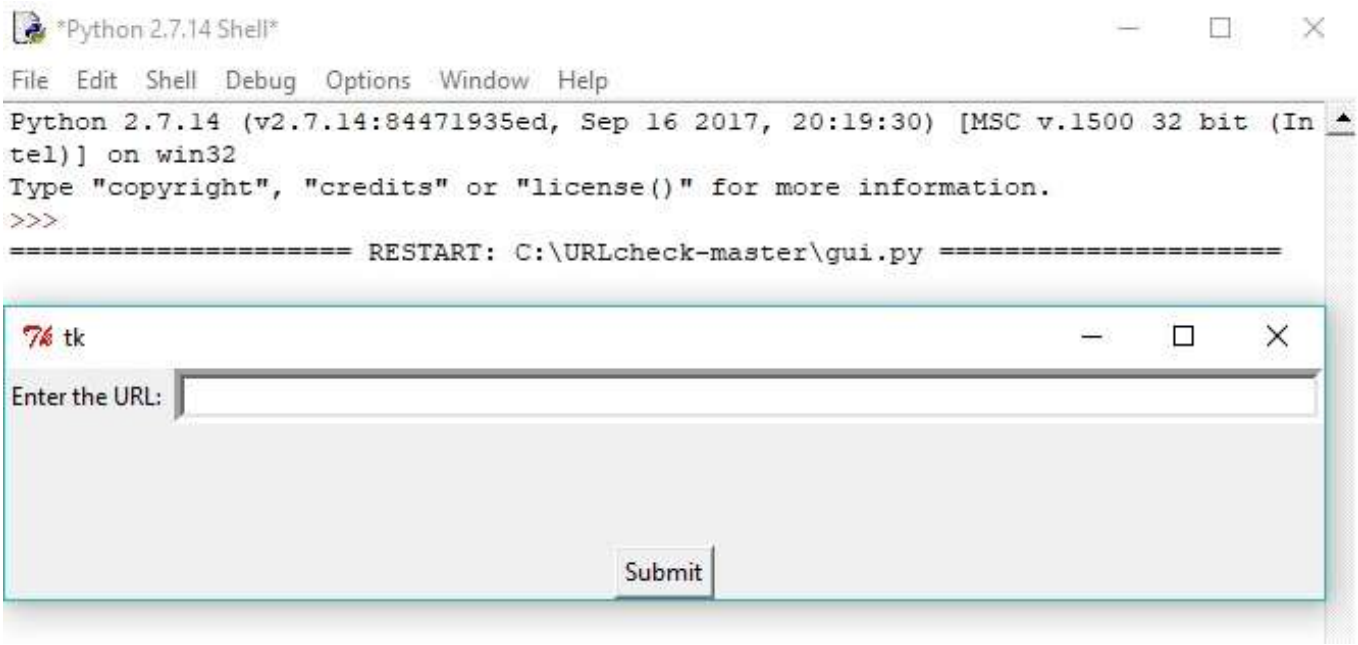


Fig.5.1: Enter the URL here

The user can give the URL here and they can check whether the URL is malicious or benign, he can give the URL in the empty text box and click on submit button and they can know the results.

After giving the individual URL in the text box and it gives a pop up message whether it is m alicious or benign.

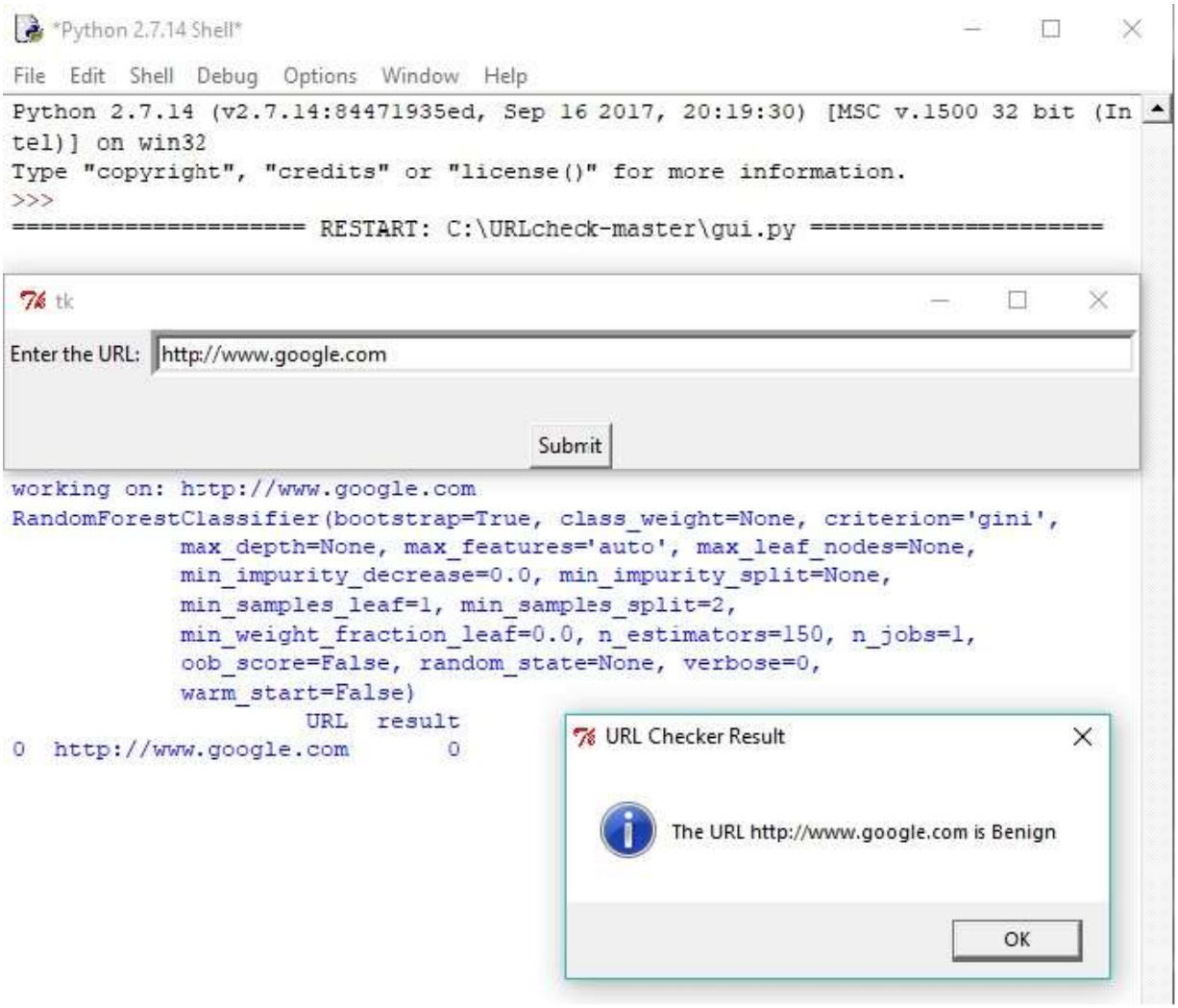


Fig 5.2: Checking the URL for benign

The above screenshot gives the result for the particular URL that is http://google.com saying t hat this particular URL http://google.com is benign, If the user gives the input and clicks the s ubmit button it will generate a pop up message as you can see in the figure which is generatin g the pop up messages. The above screenshot is for checking the Individual URL.

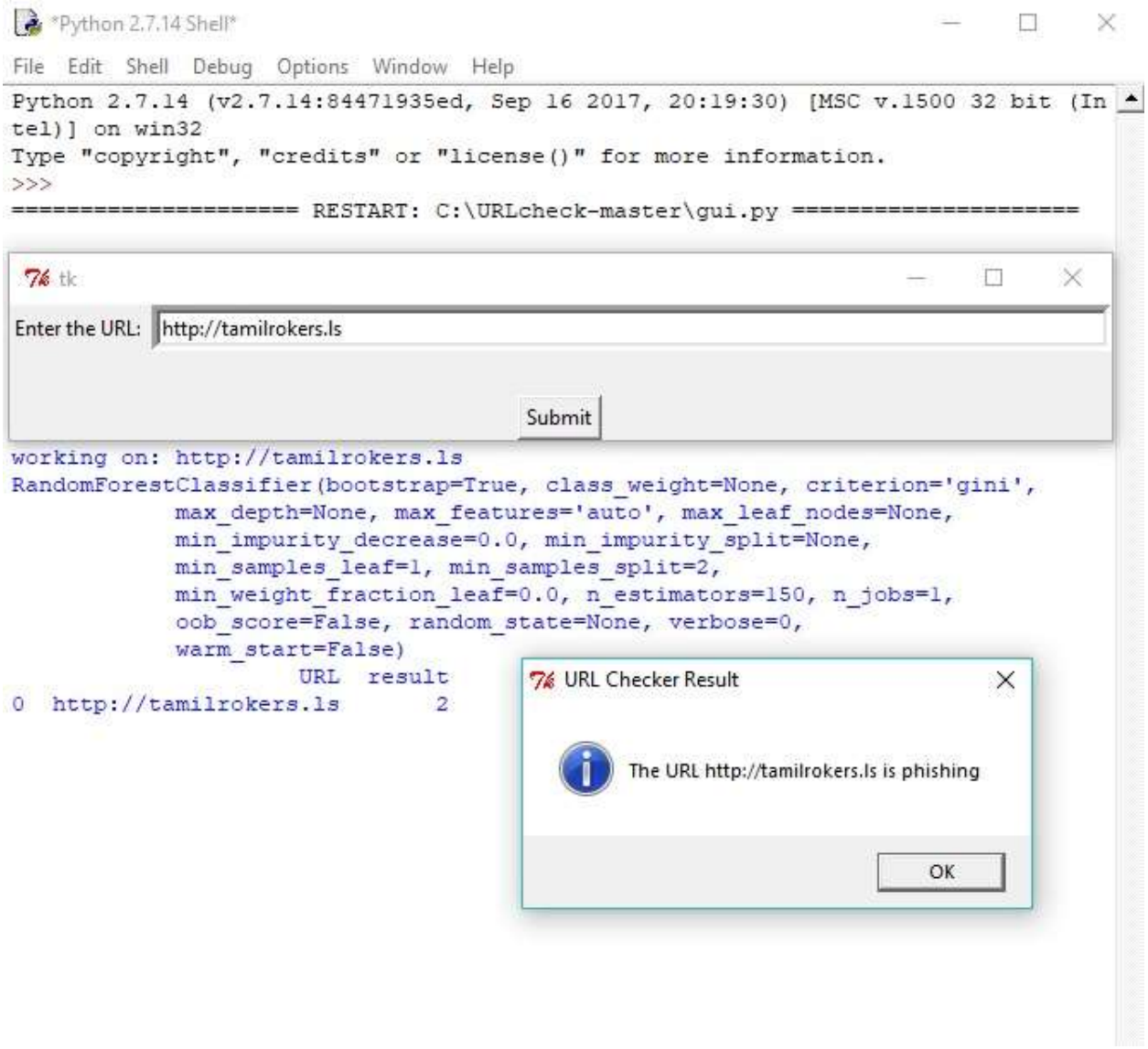


Fig 5.3: Checking the Individual URL for phishing

The above figure which is for checking for malicious URLs if the URL is said to be maliciou s then it has to classify three types of malicious URLs saying whether it is Phishing or malwa re for example I had given the URL http://youtube.com it is a malicious URL so it is displayi ng a pop up message called as the URL.

http://tamilrokers.ls is phishing URL.

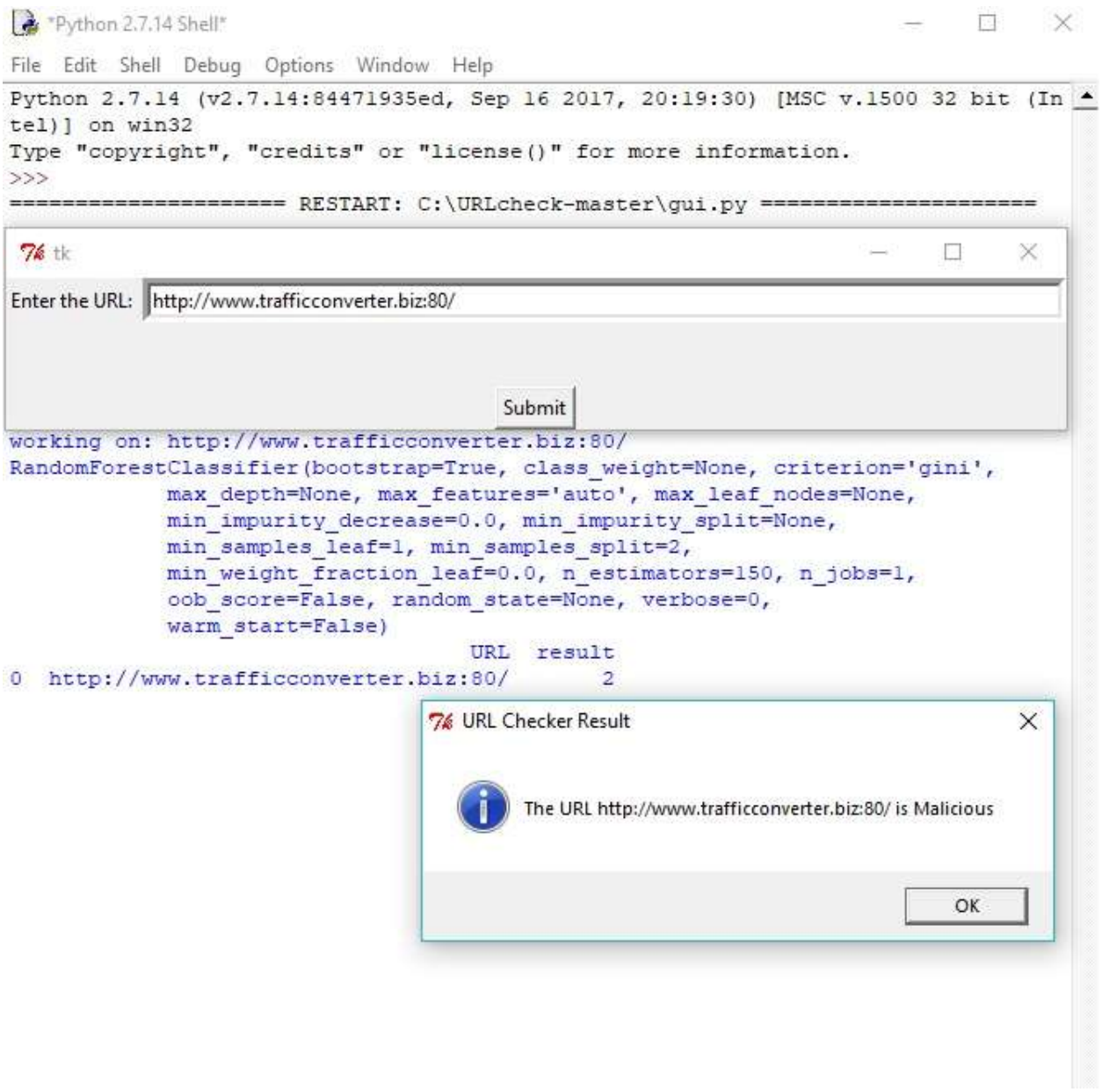


Fig 5.4: Checking the Individual URL for malware

In the above figure if the user puts the URL and clicks on the submit button and it displays a pop up message saying it is malware or malicious because it is not a benign URL, said to be a malicious URL that's why it is classifying three types of malicious URLs and the above scr eenshot is belongs to the malware class so it is said to be malicious URL.

CHAPTER 6

CODE SNIPPET

6.1 Algorithm code Snippet

```
import pandas
from sklearn import preprocessing
from sklearn.ensemble import RandomForestClassifier
import numpy
from sklearn import svm
from sklearn import cross_validation as cv
import matplotlib.pylab as plt

    def svm_classifier(train,query,train_cols):

        clf = svm.SVC()

        train[train_cols] = preprocessing.scale(train[train_cols])
        query[train_cols] = preprocessing.scale(query[train_cols])

        print clf.fit(train[train_cols], train['malicious'])
        scores = cv.cross_val_score(clf, train[train_cols], train['malicious'],
        cv=30)
        print('Estimated score SVM:%0.5f (+/- %0.5f)'%(scores.mean(),
        scores.std() / 2))
        query['result']=clf.predict(query[train_cols])
        print query[['URL','result']]
```

```
def forest_classifier_gui(train,query,train_cols):

rf = RandomForestClassifier(n_estimators=150)

print rf.fit(train[train_cols], train['malicious'])

query['result']=rf.predict(query[train_cols])

print query[['URL','result']].head(2)
return query['result']

def forest_classifier(train,query,train_cols):

rf = RandomForestClassifier(n_estimators=150)

print rf.fit(train[train_cols], train['malicious'])
scores = cv.cross_val_score(rf, train[train_cols], train['malicious'], cv=30)
print('Estimated score RandomForestClassifier: %0.5f (+/- %0.5f)' %
(scores.mean(), scores.std() / 2))

query['result']=rf.predict(query[train_cols])
print query[['URL','result']]
```

6.2 Feature extract code Snippet

```
from urlparse import urlparse
import re
import urllib2
import urllib
from xml.dom import minidom
import csv
import pygeoip

def Tokenise(url):

    if url=='':
        return [0,0,0]
    token_word=re.split('\W+',url)
    #print token_word
    no_ele=sum_len=largest=0
    for ele in token_word:
        l=len(ele)
        sum_len+=l
        if l>0:
            no_ele+=1
        if largest<l:
            largest=l
    try:
        return [float(sum_len)/no_ele,no_ele,largest]
    except:
        return [0,no_ele,largest]
```

```
def web_content_features(url):
    wfeatures={}
    total_cnt=0
    try:
        source_code = str(opener.open(url))
        #print source_code[:500]

        wfeatures['src_html_cnt']=source_code.count('<html')
        wfeatures['src_hlink_cnt']=source_code.count('<a href=')
        wfeatures['src_iframe_cnt']=source_code.count('<iframe')
        #suspicioussrc_ javascript functions count

        wfeatures['src_eval_cnt']=source_code.count('eval(')
        wfeatures['src_escape_cnt']=source_code.count('escape(')
        wfeatures['src_link_cnt']=source_code.count('link(')
        wfeatures['src_underescape_cnt']=source_code.count('underescape(')
        wfeatures['src_exec_cnt']=source_code.count('exec(')
        wfeatures['src_search_cnt']=source_code.count('search(')

        for key in wfeatures:
            if(key!='src_html_cnt' and key!='src_hlink_cnt' and
key!='src_iframe_cnt'):
                total_cnt+=wfeatures[key]
        wfeatures['src_total_jfun_cnt']=total_cnt

    except Exception, e:
        print "Error"+str(e)+" in downloading page "+url
        default_val=nf
```

```
    wfeatures['src_html_cnt']=default_val
    wfeatures['src_hlink_cnt']=default_val
    wfeatures['src_iframe_cnt']=default_val
    wfeatures['src_eval_cnt']=default_val
    wfeatures['src_escape_cnt']=default_val
    wfeatures['src_link_cnt']=default_val
    wfeatures['src_underescape_cnt']=default_val
    wfeatures['src_exec_cnt']=default_val
    wfeatures['src_search_cnt']=default_val
    wfeatures['src_total_jfun_cnt']=default_val

return wfeatures
```

```
def feature_extract(url_input):

    Feature={}
     tokens_words=re.split('\W+',url_input)       #Extract bag of words stings
    delimited by (.,/,?,,=,-,_)
    #print tokens_words,len(tokens_words)

    #token_delimit1=re.split('[./?=-_]',url_input)
    #print token_delimit1,len(token_delimit1)

    obj=urlparse(url_input)
    host=obj.netloc
    path=obj.path

    Feature['URL']=url_input

    Feature['rank_host'],Feature['rank_country'] =sitepopularity(host)

    Feature['host']=obj.netloc
    Feature['path']=obj.path

    Feature['Length_of_url']=len(url_input)
    Feature['Length_of_host']=len(host)
    Feature['No_of_dots']=url_input.count('.')

    Feature['avg_token_length'],Feature['token_count'],Feature['largest_token'] =
    Tokenise(url_input)

    Feature['avg_domain_token_length'],Feature['domain_token_count'],Feature[
    'largest_domain'] = Tokenise(host)
```

```
Feature['avg_path_token'],Feature['path_token_count'],Feature['largest_path']
= Tokenise(path)

Feature['sec_sen_word_cnt'] = Security_sensitive(tokens_words)
Feature['IPaddress_presence'] = Check_IPaddress(tokens_words)


Feature['ASNno']=getASN(host)
Feature['safebrowsing']=safebrowsing(url_input)
"""wfeatures=web_content_features(url_input)

for key in wfeatures:
   Feature[key]=wfeatures[key]

return Feature
```

CHAPTER 7

CONCLUSION

Malicious connections are military unit utilized by the aggressors to control the PC framework, which can be utilized to execute the violations, for example, spamming and phishing. The expanding levels of this wrong doings have oblige of dependable grouping and distinguishing proof of URLs. Malicious links are used by the attackers to acquire the control of computer system, The increasing level of cybercrimes have required the necessity of characterization and distinguishing proof, To recognize such violations a URL grouping and ID model is proposed in light of the Random forest classifier. From the results we can say that support vector machine model is more accurate than naive Bayes for identification and classification of malicious URLs.

REFERENCES

[1] MA, J., Saul, L. K., Savage, S., and Voelker, G. M. 2011,"Learning to Detect Malicious URLs" ACM Trans. Intell. Syst. Technol. 2, 3, Article 30 (April 2011).

[2] Anjali B. Sayamber, Arati M Dixit,"Malicious URLs detection and Identification", International journal of computer application, Volume 99 – No.17, August 2014.

[3] Hyunsang Choi.. Seoul, Bin B. Zhu."Detecting Malicious Web Links and Identifying Their Attack Types". Korea University (2011).

[4] Phishing URLs Available: Phishtank free community site for anti phishing services.

[5] Datasets are Available [Online]: http://sysnet.ucsd.edu/projects/url/url.mat.

[6] Malware datasets are Available: http://en.wikipedia.org/wiki/malware.

[7] https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

[8] http://shodhganga.inflibnet.ac.in/bitstream/10603/7989/14/14_chapter%205.pdf

[9] http://www.numpy.org/

[10] http://pandas.pydata.org/pandas-docs/stable/

[11] https://docs.scipy.org/doc/numpy-dev/user/quickstart.html

[12] http://scikit-learn.org/stable/modules/ensemble.html

[13]http://scikitlearn.org/stable/auto_examples/model_selection/plot_confusion_matri x.html

[14] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[15] http://matplotlib.org/