URL BASED DETECTION OF PHISHING WEBSITES USING MACHINE LEARNING

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

DASAM ARSHITHA (18BCD7101)

Under the Guidance of

DR. M.PRIYADHARSHINI

SCHOOL OF COMPUTER SCIENCE ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

JANUARY 2022

CERTIFICATE

This is to certify that the Capstone Project work titled "**URL BASED DETECTION OF PHISHING WEBSITES USING MACHINE LEARNING**" that is being submitted by **DASAM ARSHITHA (18BCD7101)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. M.PRIYADHARSHINI
Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

PROGRAM CHAIR
B. Tech. CSE

DEAN
School Of Computer Science Engineering

ACKNOWLEDGEMENTS

The satisfaction that accompanies the successful accomplishment of any task would be incomplete without the mention of people, who made it possible and whose guidance and encouragement crown all the effects with success.

I sincerely thank my guide **Dr.M.Priyadharshini** for the guidance and support, who has taken up several innovative projects all through the years of experience.I thank for timely suggestions and constant encouragement that boosted up our moral and to the accomplishment of the project.

It gives us immense pleasure to thank our project review committee members **Dr.Anil Carie** and **Dr.M.Priyadharshini** who have been encouraging enough to help me at time of need.

Lastly,I would like to extend my gratitude to our Vice Chancellor,**Dr.S.V.Kota Reddy** of VIT-AP University and Program Chair of CSE for providing me with all the facility that was required.Although the whole project process went through online mode due to the coronavirus pandemic,the guidance from the guide and reviews happened seamlessly.

Dasam Arshitha
(18BCD7101)

ABSTRACT

Phishing is a common type of cyber-attack these days, in which the victim is tricked into visiting bogus websites.The user can be duped into disclosing important information such as usernames, passwords, bank account information, credit card information, and so on these websites. As a result, phishers have made extensive use of it to get a user's credentials. The phishing URLs resemble the real ones in appearance but differ in some ways. Only the information about a website's URL is used in our method to identify whether the website is a phishing website or not. As a result, there is no need to visit a website in order to assess whether it is phishing or not. This also prevents the user from visiting phishing websites and being exposed to any dangerous malware they may contain. We also go through how URL meta data can be utilised to assess whether a URL is phishing or not. After that, the Random Forest technique can be used to a dataset with features like URL meta data. The random forest algorithm also has the benefit of not overfitting the data. Phishing attacks have become a significant menace to people's daily lives and the online ecosystem. In these assaults, the attacker poses as a trusted entity in order to steal sensitive data or the user's digital identity, such as account credentials, credit card numbers, and other personal information. A phishing website is a website that has a name and appearance that is similar to an official website, often known as a faked website, and is designed to deceive an individual and steal their personal information. So, in order to identify fraudulent websites, this paper will explore machine learning and deep learning algorithms and apply all of them to our dataset, with the best algorithm having the best precision and accuracy being chosen for phishing website identification.This work can provide more effective defenses for phishing attacks of the future.

KEYWORDS:

Phishing, Cyber-attacks,URL,Malicious,Random Forest, Credentials,Fraudulent.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

- Phishing is a type of cybercrime in which the perpetrators take advantage of the fact that it is simple, inexpensive, and effective.
- Fraudsters try to get sensitive and personal information such as passwords, usernames, and bank details such as credit/debit card numbers by impersonating a trustworthy business through electronic communication.
- The phishing website will look just like the authentic website and will drive the user to a page on the fraudulent website where they may submit their personal information.
- The suggested system detects phishing URLs using any of the available machine learning methods, forecasts their correctness, and finds the phishing website.

Fig1 Beware of Phishing

1.1 Objectives

Detecting phishy URLs is crucial to protect users privacy and brand defame, users private information unauthorized utilization is crime as per data privacy policy .the first case of mass targeting and stealing information noted for American online website similarly many cases been occurring until now, There are major approaches in identifying phishing sites many researchers developed different methodologies such as Delta-phish ,Det-phish, phish-safe, phishDef systems and also approached analyzing the css syntax as effective features of web page and user behaviors such as Human Interaction Proofs concept and some works on server side methodology by using natural language processing(NLP) and machine learning (ML) with probabilistic model to detect phishy nature.

1.2 Background and Literature Survey

The current scenario is that the vast majority of the public has been duped into providing personal information to hackers without their knowledge. Many blacklisted websites have been published to appear as an original site in order to trap user by asking them to input their personal details. For instance, a password, a bank account, an email address, and so forth. Phishing activity in early 2016 was at an all-time high since the company began tracking it in 2004. In 2016, a total of 1,220,523 phishing attacks were reported. In comparison to 2015, this was a 65 percent increase. There were 1,609 phishing assaults per month in the fourth quarter of 2004. There were 92,564 phishing assaults every month on average in the fourth quarter of 2016, up 5,753 percent over the previous twelve months. According to the Anti-Phishing Working Group (APWG), at least 47, 324 phishing attacks occur each year, with a top-ten American bank estimating that at least US$300 is lost for every hour a phishing site is active. The science of getting computers to act while not being expressly programmed is known as machine learning. Machine Learning was implemented to develop this proposed system. Machine learning techniques identifies phishing URLs typically assess a URL based on some feature or set of features extracted from it.Thus, before coming to conclusion that this was the major problem, related products were examined and compared view their libation before progressing to the proposed project..Phishing is a type of web-based assault that persuades consumers to visit fake websites and reveal sensitive information such as their user id and password. Phishing webpages are created by impostors who replicate a web page from a legitimate source. These fraudulent web pages look identical to the legitimate ones. To address this issue, this project proposes a machine learning-based anti-phishing technique that extracts features and only gives the user information. We investigated the numerous characteristics of phishing and authentic websites in depth and identified features that can be used to distinguish phishing from valid websites. Protecting users from phishing attacks and fake websites is an important aspect of online security. Intelligent methods can be used to develop fake webpages.Phishing attacks can be launched via file sharing, blogs, and forums can be used by attackers for phishing. Legal measures, education, and technology solutions are all available to combat phishing.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology, ML Algorithms used, System Analysis, Application Modules, System Design details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

2. Proposed System

In order to create a practical development environment and a realistic schedule, a project methodology must be chosen and defined as part of the effort to develop the proposed system. As a result, for its shortened development period and flexible process, this project will be completed using the Agile Unified Process (AUP) Lifecycle. A collection of hardware and software requirements is established as a starting point. This is to ensure that the operating system platform can handle and carry out the system's development. The software that is used to develop system is using Java NetBeansIDE and Jupyter notebook.

- To train the machine learning algorithms on the dataset and compare the obtained results for trained models and specify which is best.
- With the help of the specified best algorithm,we detect the phishing websites URLs.
- By using a web application,we check whether the provided URL belongs to a Phishing Website or not.
- Main aim is to ensure knowledge and security for the users as they are the victims of these phishing websites.

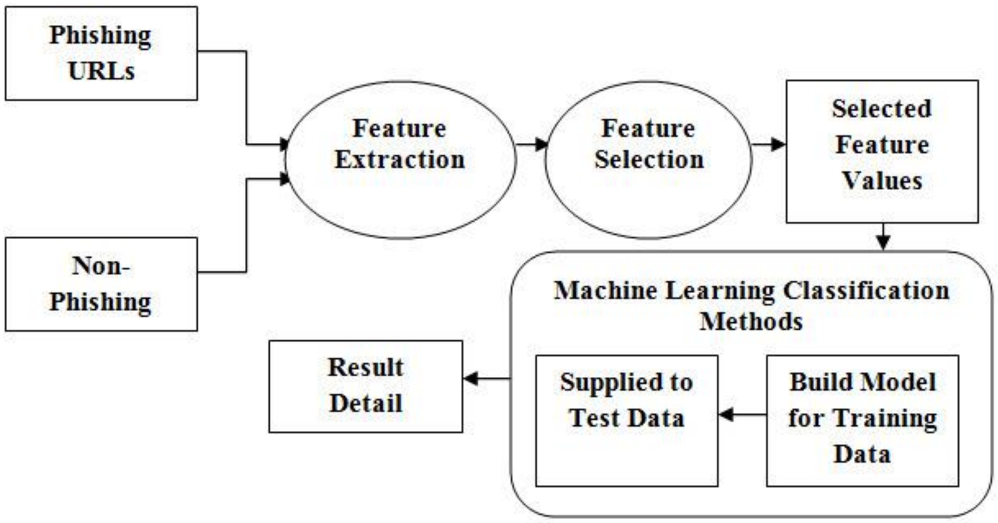


Fig2 Proposed System Block Diagram

2.1. Working Methodology

The system has two sections, hardware and software. The project consists of a laptop or a computer having installed with Anaconda navigator(Jupyter notebook) and NetBeansIDE.

Phase-1: Collecting a Dataset suitable for detecting the phishing websites based on their URLs.

Phase-2: Testing the dataset with ML algorithms like Decision Tree,Support Vector Machine and Random Forest with the help of Jupyter Notebook and obtaining the best algorithm suitable for the project.

Phase-3: Using a web application, we provide an URL to detect whether it is a phishing website or not.

Phase-4: Tuning the URL with the obtained best Machine Learning algorithm.

Phase-5: Displaying whether it is a Phishing website or not.

2.2. Machine Learning Algorithms

ML Algorithms used in this project are:

- **Random Forest Classifier**

A Random Forest is an ensemble technique that uses several decision trees with a technique called Bootstrap Aggregation, sometimes known as bagging, to solve both regression and classification problems. Instead of depending on individual decision trees, the main idea is to aggregate numerous decision trees to determine the final outcome.

• Choose K data points at random from the training set.

• Create a decision tree for each of the K data points.

• Repeat steps 1 and 2 for the number Ntree of trees you want to build.

• Make each of your Ntree trees estimate the value of Y for a new data point, then assign the new data point the average of all of the anticipated Y values.

- **Decision Tree Classifier**

The decision tree starts by selecting the best splitter from the available qualities for categorization, which is referred to as the tree's root. The tree is built by the algorithm till it reaches the leaf node. In tree representation, each internal node of the tree corresponds to attribute and each leaf node of the tree belongs to class label, which is used to anticipate target value or class. The decision tree is a dispense-free or non-parametric method, which does not rely on probability distribution inference. They are capable of managing large amounts of dimensional data with the utmost precision.It easily identifies non-linear patterns and it also requires only a few data preprocessing, in other words, there is no necessity to normalize columns. It is also used for feature engineering such like finding missing data.

- **Support Vector Machines**

  One of the most widely used classifiers is support vector machines (SVMs). The goal of SVM is to find the point where two classes are the closest by using the maximum distance between them. Support vector machines (SVMs, also known as support vector networks) are supervised learning models with related learning algorithms for classification and regression analysis in machine learning. A Support Vector Machine (SVM) is a discriminative classifier with a separating hyperplane as its formal specification.In other words, the algorithm produces an ideal hyperplane that categorizes fresh samples given labelled training data (supervised learning). An SVM model is a representation of the instances as points in space that have been mapped to separate the examples of the various categories by a significant distance. SVMs may also conduct non-linear classification, implicitly translating their inputs into high-dimensional feature spaces, in addition to linear classification.

2.3. System Analysis

2.3.1 Software Requirements

Operating System	Only for Windows 7 & above
Frontend	JSP
Language	Java & Python
IDE	Anaconda Navigator,NetBeans
Platform	JupyterNotebook,NetBeansIDE8.2,Web Browser

Table 1 Software Requirements

1. **Anaconda Navigator-Jupyter Notebook :**

***Anaconda Navigator*_**is a Python and R programming language distribution for scientific computing that seeks to make package management and deployment easier. Data-science packages for Windows, Linux, and macOS are included in the release. Anaconda Navigator is a desktop graphical user interface (GUI) included with the Anaconda distribution that allows users to run programmes and manage conda packages, environments, and channels without using command-line commands.

Navigator comes with the following applications pre-installed:

- JupyterLab
- Jupyter Notebook
- QtConsole[19]
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computing platform for creating notebook papers. A Jupyter Notebook document is an ordered list of input/output cells that can contain code, text (Markdown), mathematics, graphs, and rich media.A notebook is a JSON document that follows a versioned format and commonly ends with the ".ipynb" extension beneath the interface. Jupyter Notebook can connect to a variety of kernels, allowing you to programme in a variety of languages. A Jupyter kernel is a programme that handles a variety of requests (code execution, code completions, and inspection) and responds.

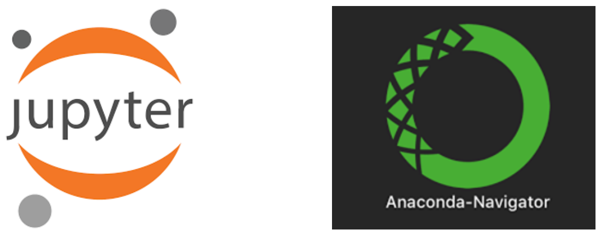


Fig3 JupyterNotebook & Anaconda

2. **NetBeansIDE:**

NetBeans is a Java-based integrated development environment (IDE). NetBeans enables the creation of applications from a set of modular software components known as modules. NetBeans is compatible with Windows, Mac OS X, Linux, and Solaris. It has extensions for PHP, C, C++, HTML5, and JavaScript, in addition to Java development. Third-party developers can extend NetBeans-based applications, including the NetBeans IDE.

Fig4 NetBeansIDE

3. **Python:**

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasises code readability by using a lot of indentation. Its language features and object-oriented approach are designed to help programmers write clear, logical code for both small and large-scale projects.

Fig5 Python

4. **Java:**

Java is a high-abstraction object-oriented programming language with as few implementation dependencies as feasible. It's a general-purpose programming language that allows programmers to write once and run anywhere (WORA), which implies that compiled Java code may run on any platform that supports Java without needing to be recompiled. Java programmes are often compiled to bytecode, which may run on any Java virtual machine (JVM) regardless of computer architecture.

Fig6 Java

5. **Java Server Pages:**

Jakarta Server Pages (JSP; formerly JavaServer Pages) is a set of technologies that enable software developers to build dynamically produced web pages using HTML, XML, SOAP, and other document types. Sun Microsystems released JSP in 1999, which is comparable to PHP and ASP but uses the Java programming language.

A suitable web server with a servlet container, such as Apache Tomcat, Jetty, or NetBeans, is required to deploy and run Jakarta Server Pages.

Fig7 JSP

2.3.2 Hardware Requirements

Processor	Any processor above 500 MHz
HardDisk	1TB
RAM	4 GB(minimum)

Table 2 Hardware Requirements

1. **Processor :**

A ***processor***, often known as a processing unit, is a digital circuit that conducts operations on data from an external source, usually memory or another data stream.It is usually in the form of a microprocessor, which can be implemented on a single metal–oxide–semiconductor integrated circuit chip; formerly, processors were built using several discrete vacuum tubes, transistors, or integrated circuits.

Fig8 Processor

2. **HardDisk:**

A ***hard disc*** drive (HDD), also known as a hard disc, hard drive, or fixed disc, is an electro-mechanical data storage device that uses magnetic storage and one or more rigid quickly rotating platters coated with magnetic material to store and retrieve digital data.

Fig9 HardDisk

3. **RAM:**

Random-access memory (**RAM**) is a type of computer memory that can be read and altered in any sequence and is commonly used to store working data and machine code. A random-access memory device allows the data objects to be read or written in about the same amount of time regardless of where they are physically located within the memory.

Fig10 RAM

2.4. Application Modules

User Login Page	: It is the starting page.Here the user can LogIn using his credentials and can have access to our web application.
HOME page	: It briefly explains the information related to the phishing websites so, if any user have doubts regarding their synonyms they can have small reading.
ML Algorithm	
Results Page	: It displays the results of tuning Random Forest Classifier on the dataset. It clearly explains the scores obtained by the algorithm.
Detect URL Here	: The Detection of URL's whether it is a phished website or not,Spam website or not, contains Malware or not,Malicious website or not.
Log Out page	: It safely takes you to the user login page.

2.5. System Design

Introduction to UML(Unified Modeling Language) :

The functionality supplied by a system to external integrators is depicted in these diagrams. Actors, use cases, and their relationships are depicted in these diagrams. UML is a method for using the blue print to describe the system architecture in detail. UML is a set of best engineering practises for designing large and complex systems that has been demonstrated to work.
The following are the most common types of UML diagrams:

1.Use Case Diagram
2.Class Diagram
3.Sequence Diagram
4.Activity Diagram

Use Case Diagram

The system's functionality is represented by a use case diagram. The use case focuses on the system's behaviour from the outside. External entities that engage with the system are referred to as actors.

Use cases: A use case is a horizontal ellipse that represents a sequence of actions that deliver something of measurable value to an actor.

Actors: A person, organisation, or external system that participates in one or more interactions with the system is referred to as an actor.

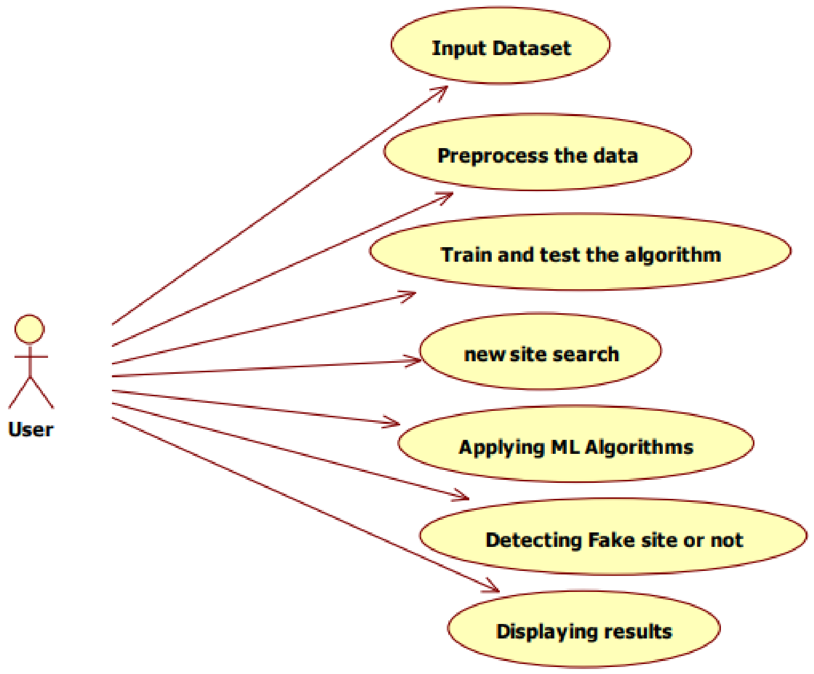


Fig11 Use Case Diagram

Class Diagram

Class-based Modeling, also known as class-oriented programming, is a type of object-oriented programming in which inheritance is accomplished by specifying classes of objects rather than the objects themselves. A class-based form of OOP, as opposed to an object-based one, is the most popular and developed. Objects are entities in this model that combine state (data), behaviour, and identity. A class is a definition, or blueprint, of all objects of a certain kind, and it defines the structure and behaviour of that item. An object must be generated expressly based on a class, and any object produced in this manner is considered an instance of that class. Method pointers, member access control, and an implicit data member that locates instances of the class in the class hierarchy distinguish an object from a structure.

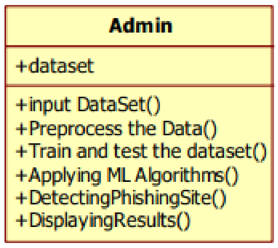


Fig12 Class Diagram

Sequence Diagram

A sequence diagram is a type of interaction diagram that displays how and in what order processes interact with one another. It's a Message Sequence Chart construct. Event diagrams, event scenarios, and timing diagrams are all terms used to describe sequence diagrams. A sequence diagram depicts multiple processes or things that exist simultaneously as parallel vertical lines (lifelines), and the messages passed between them as horizontal arrows, in the order in which they occur. This enables for the graphical specification of simple runtime scenarios. It displays a role if the lifeline is that of an object. It's worth noting that leaving the instance name blank can be used to represent anonymous or unidentified instances. Messages are used to demonstrate interaction. The message's name is written above these horizontal arrows. Asynchronous calls are represented by solid arrows with full heads, asynchronous calls by solid arrows with stick heads, and return messages by dashed arrows with stick heads.

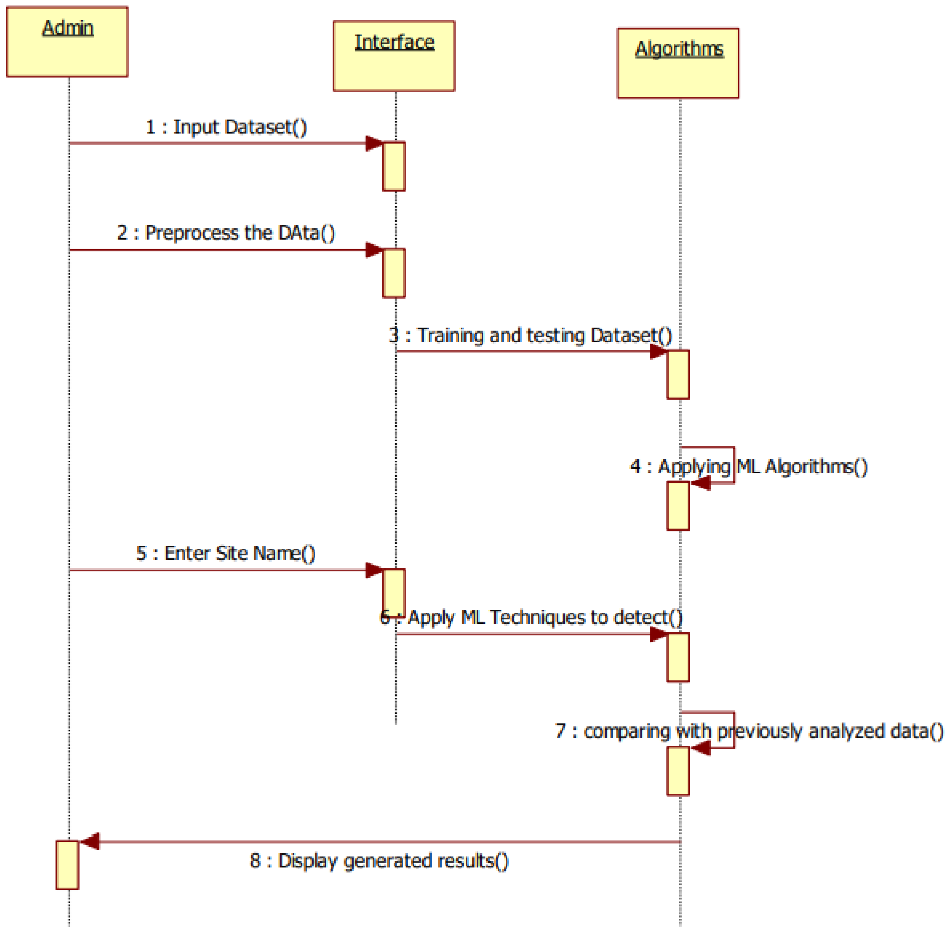

Admin
Interface
Algorithms
1 : Input Dataset()
2 : Preprocess the DAta()
3 : Training and testing Dataset()
4 : Applying ML Algorithms()
5 : Enter Site Name()
6 : Apply ML Techniques to detect()
7 : comparing with previously analyzed data()
8 : Display generated results()

Fig13 Sequence Diagram

Activity Diagram

Activity diagrams are graphical representations of workflows with support for choice, iteration, and concurrency in stepwise activities and actions. Activity diagrams can be used in the Unified Modeling Language to depict the business and operational step-by-step processes of system components. An activity diagram depicts the total control flow. Activity diagrams are made up of a restricted number of shapes that are connected by arrows.
The most common shape kinds are:

1. Activities are represented by circular rectangles.
2. Diamonds are symbolic of choices.
3. The start (split) or end (join) of concurrent actions are represented by bars.
4. The start (starting state) of the workflow is represented by a black circle.
5. The end is symbolised by a black circle that is encircled (final state).
6. The order in which actions occur is represented by arrows that run from the beginning to the conclusion.

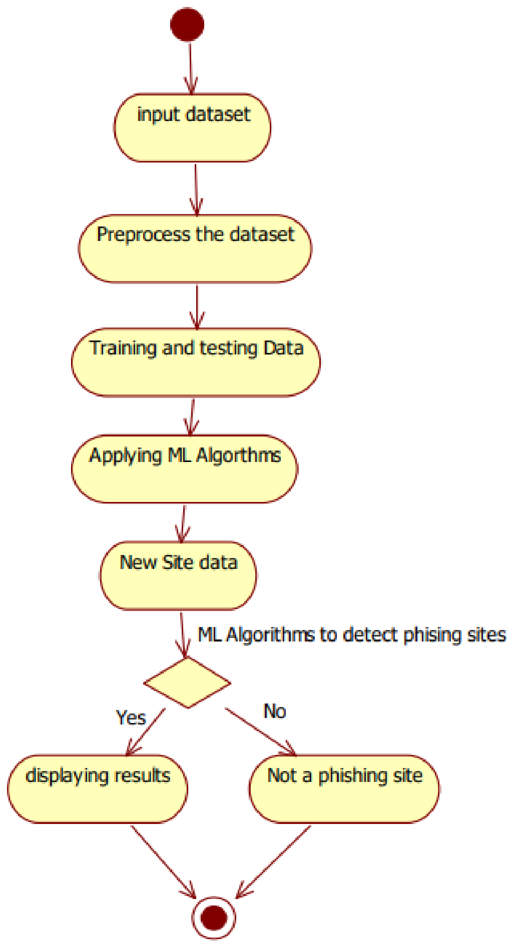


Fig14 Activity Diagram

CHAPTER 3
COST ANALYSIS

3.1. List of components and their cost

The various components used in this project are all softwares that can be installed through internet for free.

S.No		
1.	**LOC – Server Side**	**222**
2.	**LOC – Model Deployment**	**219**
3.	**LOC – Model Justification**	**150**
4.	**Number of people**	**1**
5.	**Number of man hours per day**	**~ 4 hrs**

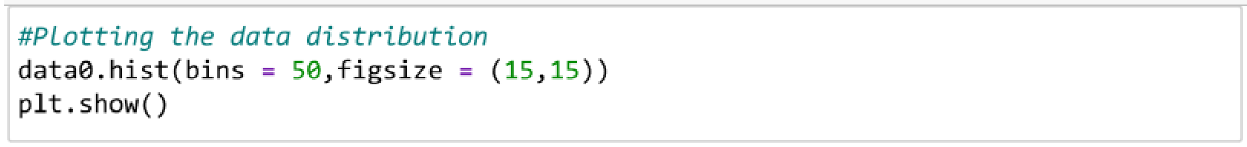
LOC-Lines Of Code

Table 3 List of components and their costs

CHAPTER 4

RESULTS AND DISCUSSIONS

Histograms of all the columns of dataset :



```
#Plotting the data distribution
data0.hist(bins = 50,figsize = (15,15))
plt.show()
```

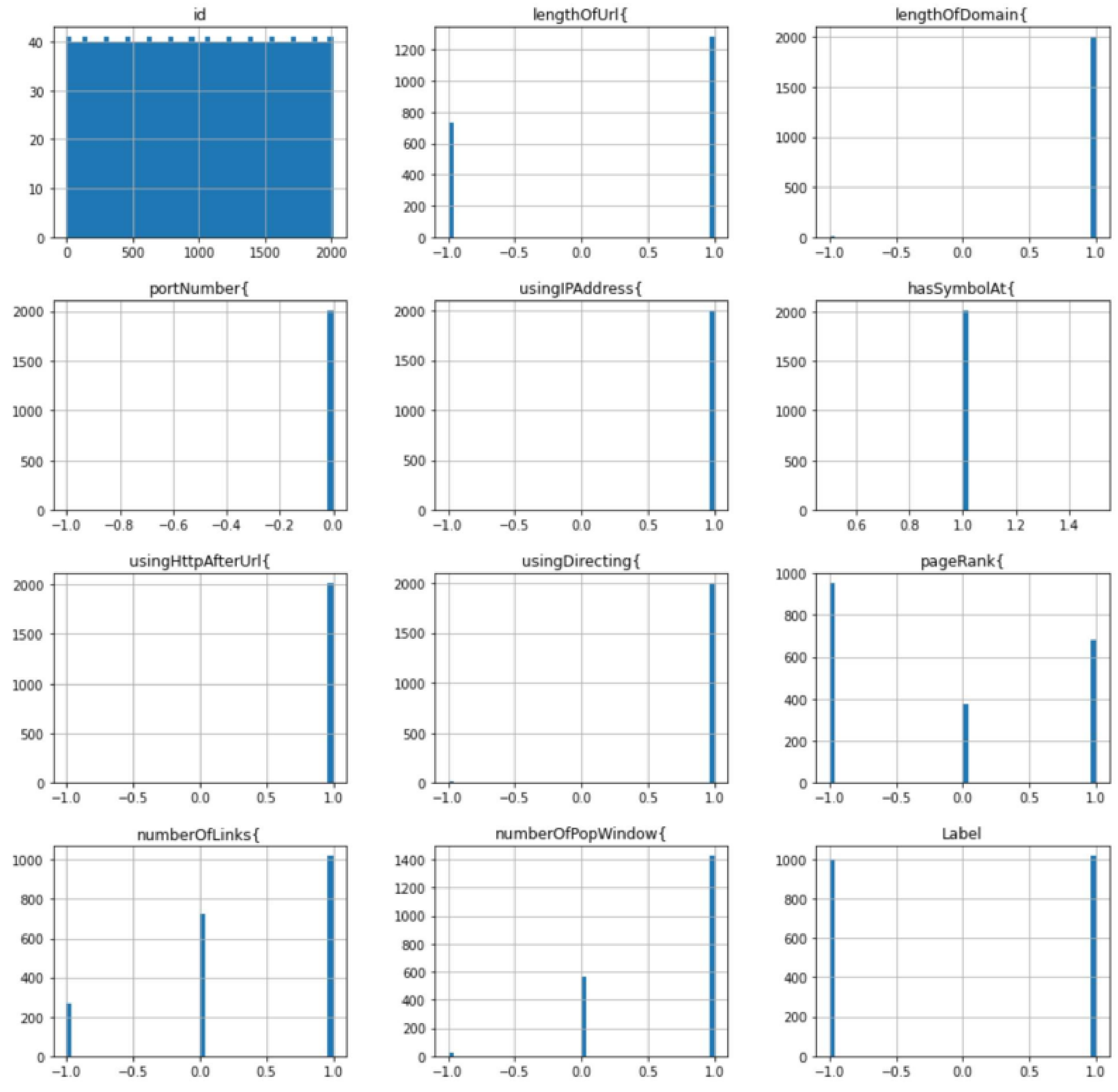
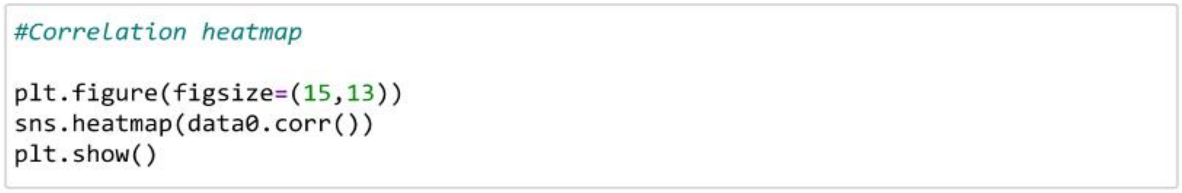


Fig15 Histogram

Correlation Matrix of Dataset :



```
#Correlation heatmap

plt.figure(figsize=(15,13))
sns.heatmap(data0.corr())
plt.show()
```

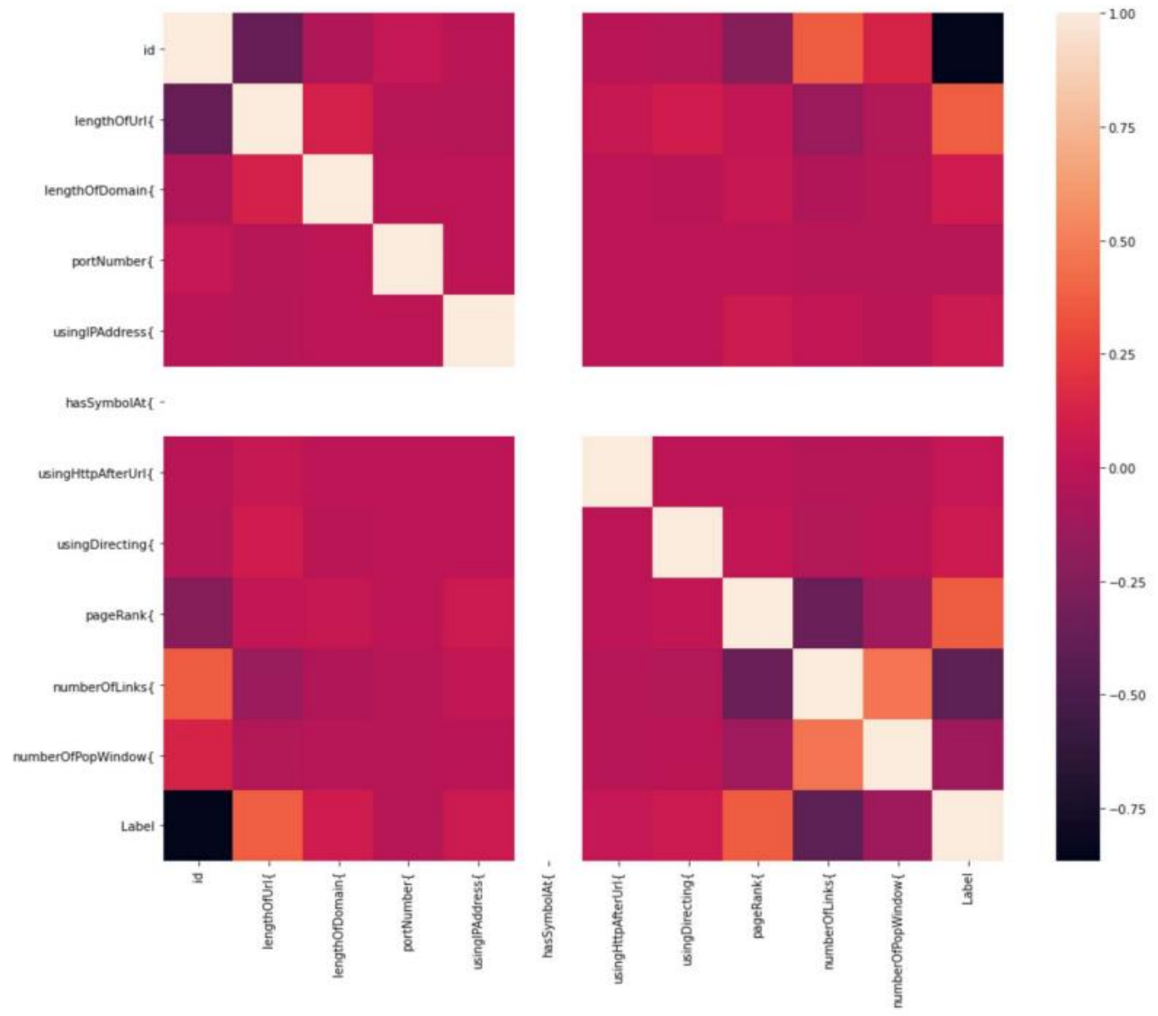
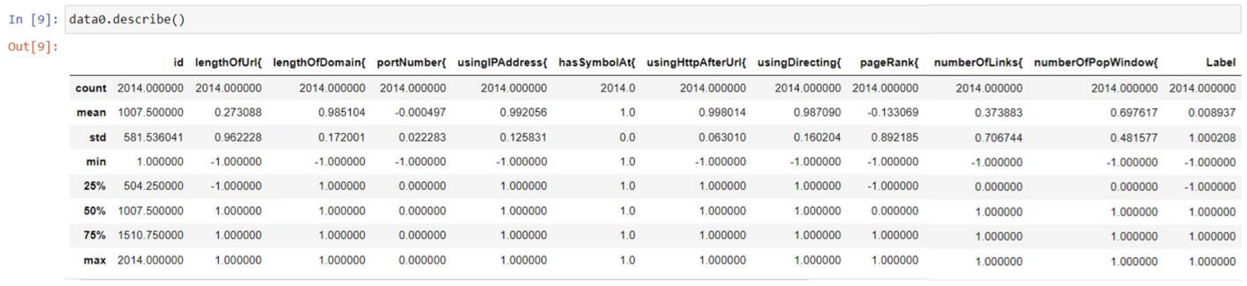


Fig16 Correlation Matrix

Description of Dataset:

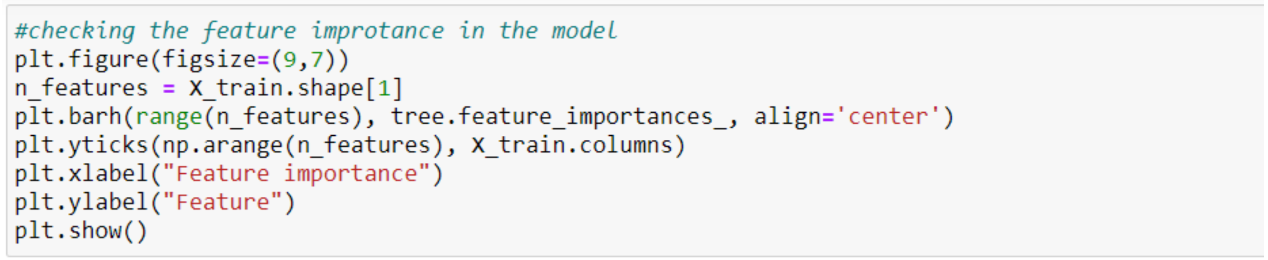


```
In [9]: data0.describe()
Out[9]:
```

	id	lengthOfUrl{	lengthOfDomain{	portNumber{	usingIPAddress{	hasSymbolAt{	usingHttpAfterUrl{	usingDirecting{	pageRank{	numberOfLinks{	numberOfPopWindow{	Label
count	2014.000000	2014.000000	2014.000000	2014.000000	2014.000000	2014.0	2014.000000	2014.000000	2014.000000	2014.000000	2014.000000	2014.000000
mean	1007.500000	0.273088	0.985104	-0.000497	0.992056	1.0	0.998014	0.987090	-0.133069	0.373883	0.697617	0.008937
std	581.536041	0.962228	0.172001	0.022283	0.125831	0.0	0.063010	0.160204	0.892185	0.706744	0.481577	1.000208
min	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	1.0	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
25%	504.250000	-1.000000	1.000000	0.000000	1.000000	1.0	1.000000	1.000000	-1.000000	0.000000	0.000000	-1.000000
50%	1007.500000	1.000000	1.000000	0.000000	1.000000	1.0	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000
75%	1510.750000	1.000000	1.000000	0.000000	1.000000	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	2014.000000	1.000000	1.000000	0.000000	1.000000	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig17 Description of Dataset

Decision Tree Performance Analysis :



```
#checking the feature improtance in the model
plt.figure(figsize=(9,7))
n_features = X_train.shape[1]
plt.barh(range(n_features), tree.feature_importances_, align='center')
plt.yticks(np.arange(n_features), X_train.columns)
plt.xlabel("Feature importance")
plt.ylabel("Feature")
plt.show()
```

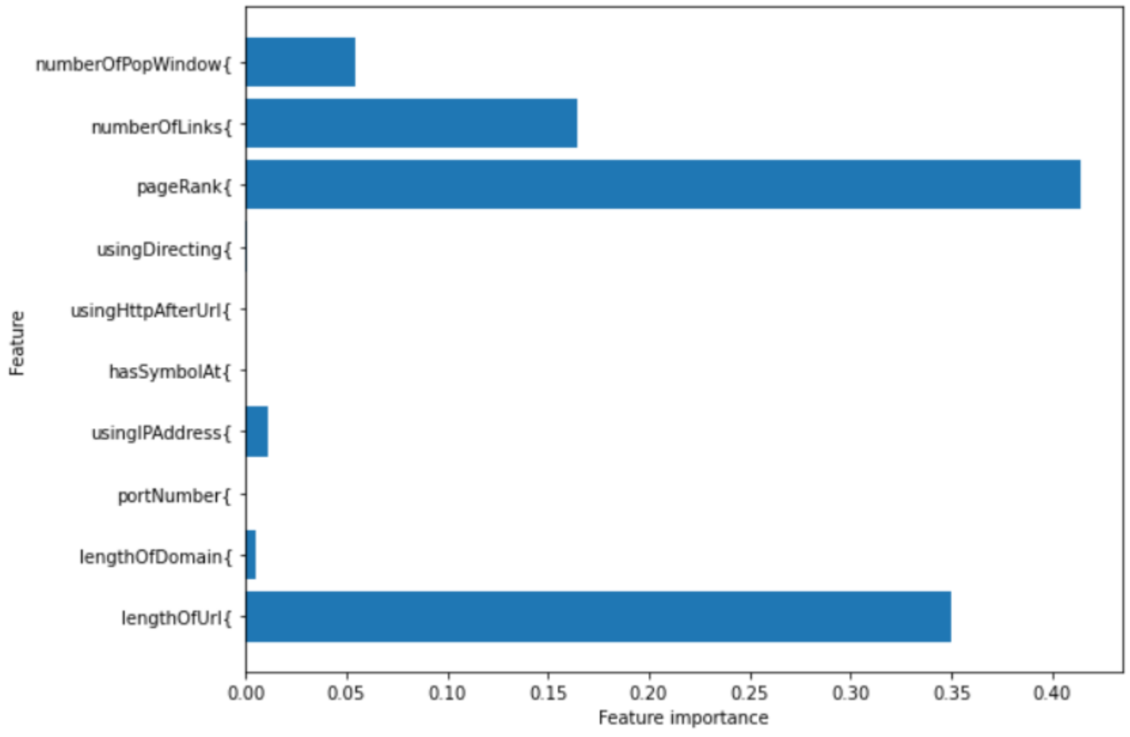
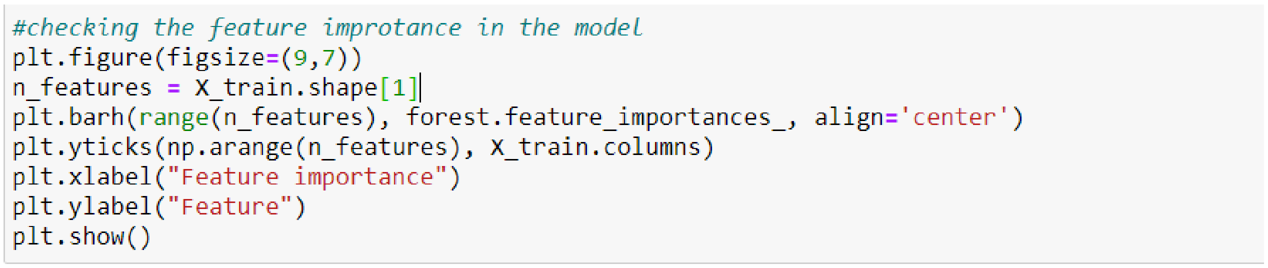


Fig18 Decision Tree Performance Analysis

Random Forest Classifier Performance Analysis :



```
#checking the feature improtance in the model
plt.figure(figsize=(9,7))
n_features = X_train.shape[1]
plt.barh(range(n_features), forest.feature_importances_, align='center')
plt.yticks(np.arange(n_features), X_train.columns)
plt.xlabel("Feature importance")
plt.ylabel("Feature")
plt.show()
```

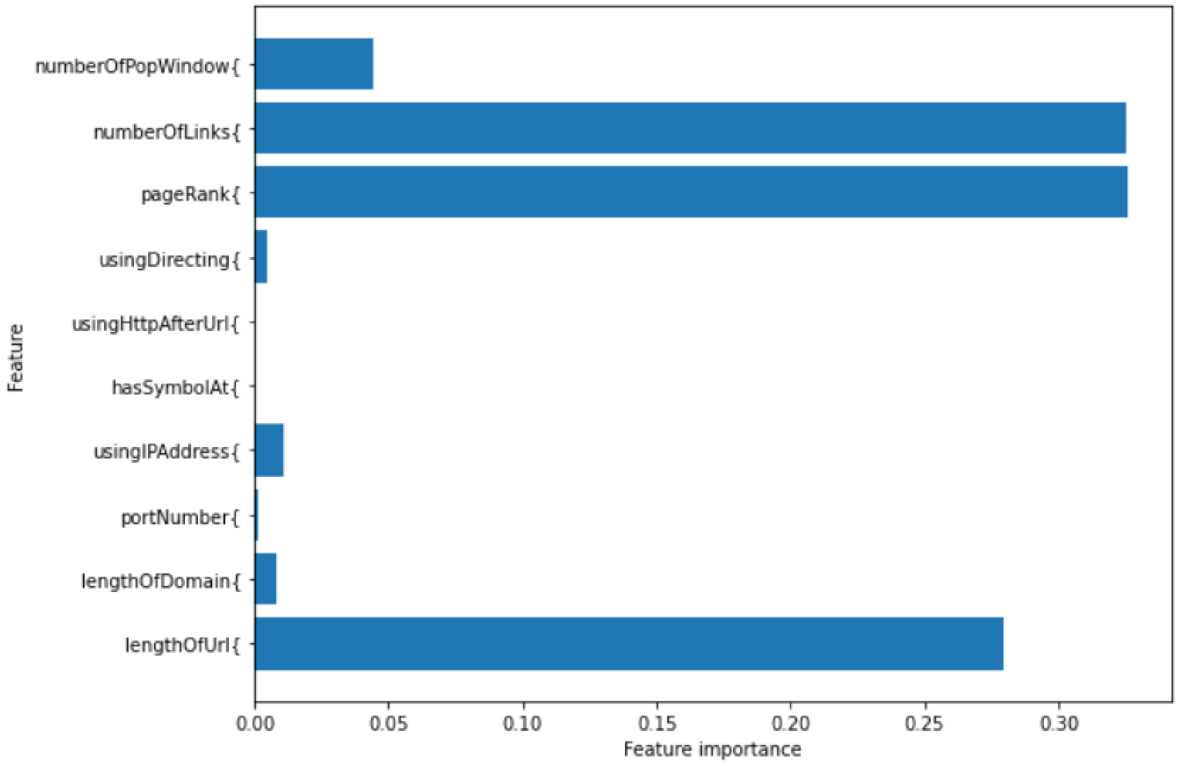
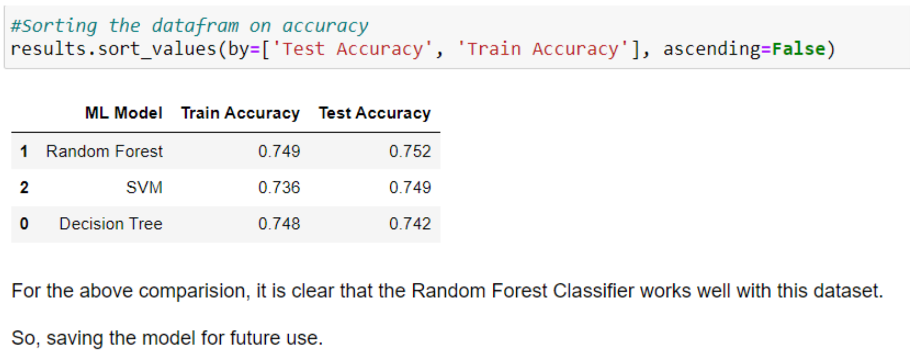


Fig19 Random Forest Classifier Performance Analysis

Comparison Results:



```
#Sorting the datafram on accuracy
results.sort_values(by=['Test Accuracy', 'Train Accuracy'], ascending=False)
```

	ML Model	Train Accuracy	Test Accuracy
1	Random Forest	0.749	0.752
2	SVM	0.736	0.749
0	Decision Tree	0.748	0.742

For the above comparision, it is clear that the Random Forest Classifier works well with this dataset.

So, saving the model for future use.

Fig20 Comparison Results

Admin Login Page: If the valid credentials are given, it takes us into the home page otherwise it stays on the same page.

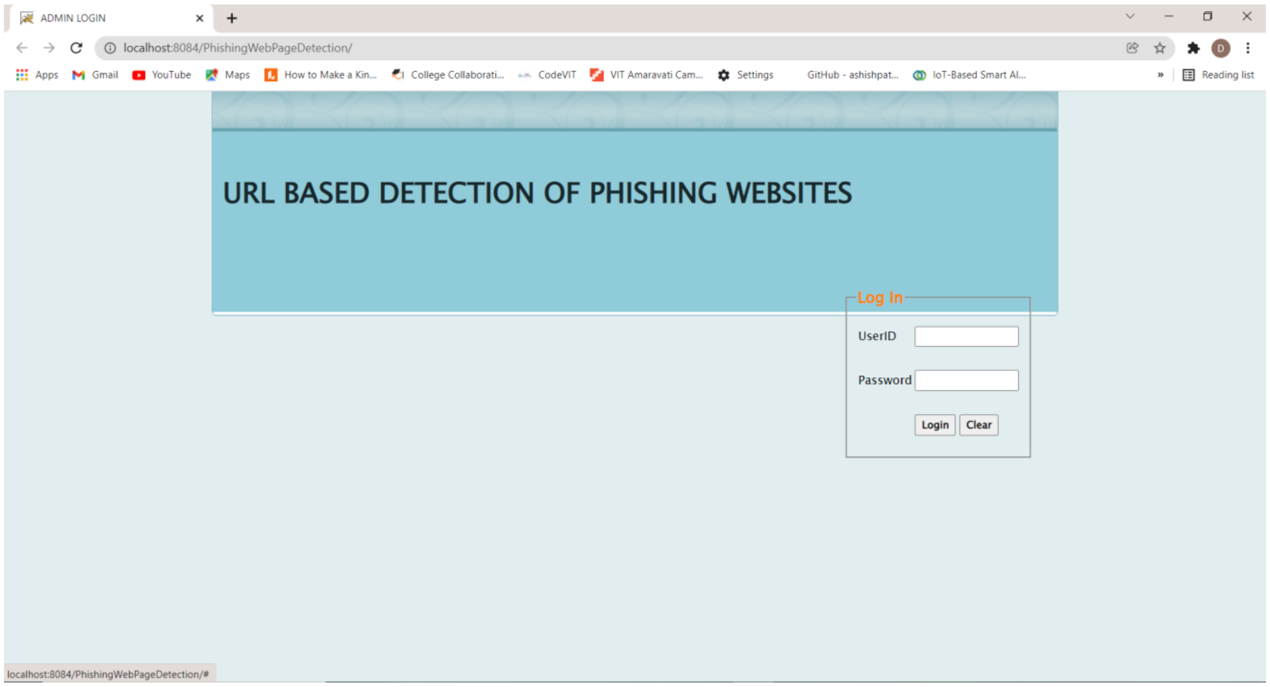
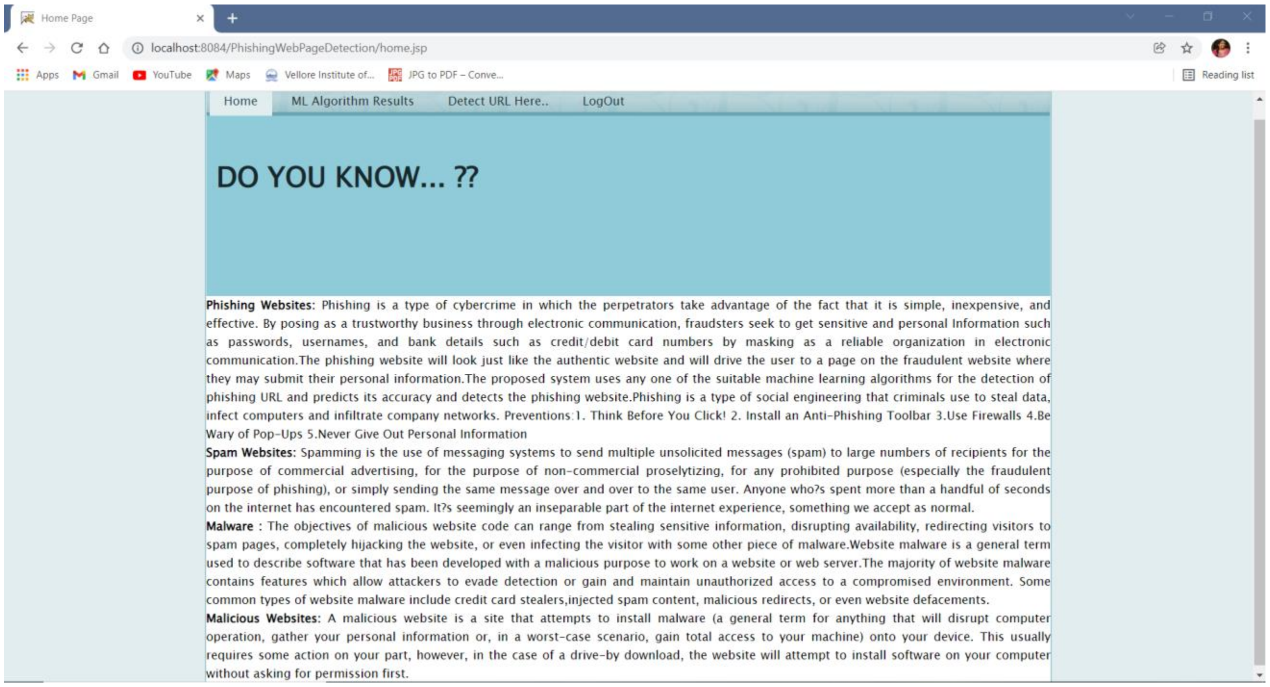

ADMIN LOGIN
localhost:8084/PhishingWebPageDetection/
URL BASED DETECTION OF PHISHING WEBSITES
Log In
UserID
Password
Login
Clear
localhost:8084/PhishingWebPageDetection/#

Fig21 Admin Login page

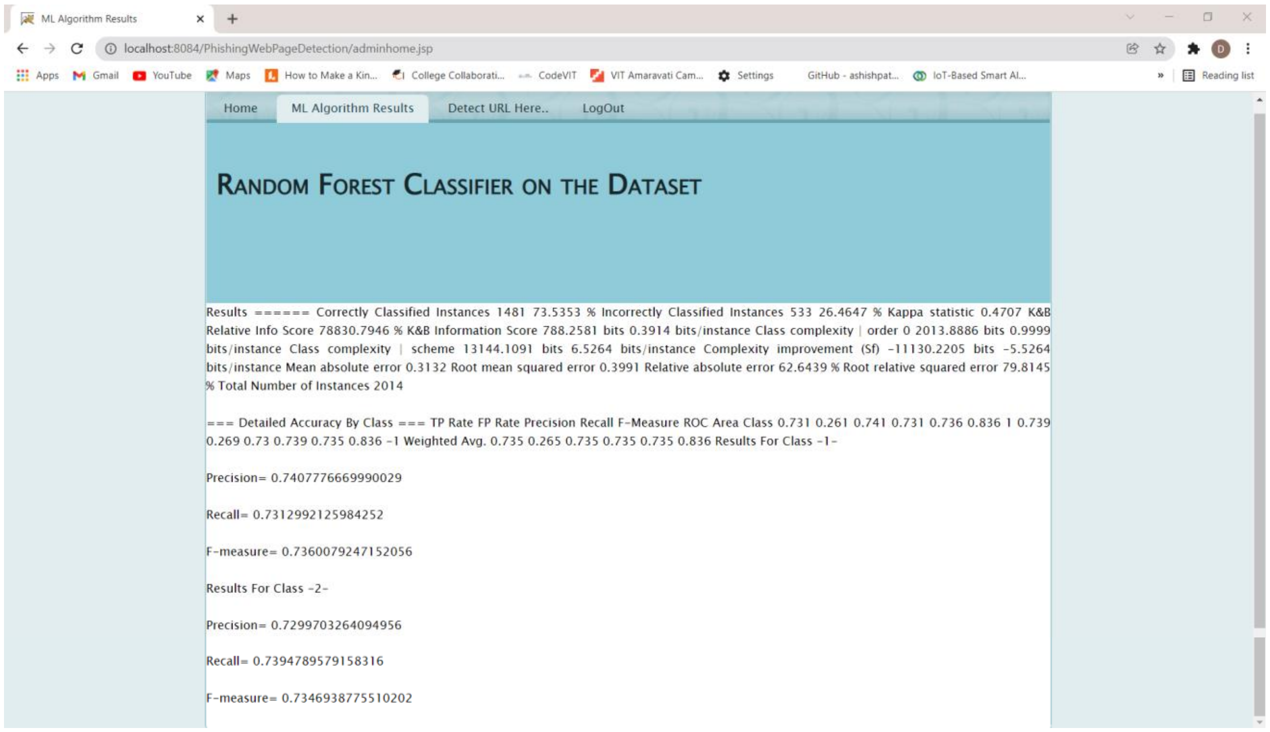
Home Page: It has brief description of the phishing websites.So that one can have a glance in need.



Phishing Websites: Phishing is a type of cybercrime in which the perpetrators take advantage of the fact that it is simple, inexpensive, and effective. By posing as a trustworthy business through electronic communication, fraudsters seek to get sensitive and personal Information such as passwords, usernames, and bank details such as credit/debit card numbers by masking as a reliable organization in electronic communication.The phishing website will look just like the authentic website and will drive the user to a page on the fraudulent website where they may submit their personal information.The proposed system uses any one of the suitable machine learning algorithms for the detection of phishing URL and predicts its accuracy and detects the phishing website.Phishing is a type of social engineering that criminals use to steal data, infect computers and infiltrate company networks. Preventions:1. Think Before You Click! 2. Install an Anti-Phishing Toolbar 3.Use Firewalls 4.Be Wary of Pop-Ups 5.Never Give Out Personal Information
Spam Websites: Spamming is the use of messaging systems to send multiple unsolicited messages (spam) to large numbers of recipients for the purpose of commercial advertising, for the purpose of non-commercial proselytizing, for any prohibited purpose (especially the fraudulent purpose of phishing), or simply sending the same message over and over to the same user. Anyone who?s spent more than a handful of seconds on the internet has encountered spam. It?s seemingly an inseparable part of the internet experience, something we accept as normal.
Malware : The objectives of malicious website code can range from stealing sensitive information, disrupting availability, redirecting visitors to spam pages, completely hijacking the website, or even infecting the visitor with some other piece of malware.Website malware is a general term used to describe software that has been developed with a malicious purpose to work on a website or web server.The majority of website malware contains features which allow attackers to evade detection or gain and maintain unauthorized access to a compromised environment. Some common types of website malware include credit card stealers,injected spam content, malicious redirects, or even website defacements.
Malicious Websites: A malicious website is a site that attempts to install malware (a general term for anything that will disrupt computer operation, gather your personal information or, in a worst-case scenario, gain total access to your machine) onto your device. This usually requires some action on your part, however, in the case of a drive-by download, the website will attempt to install software on your computer without asking for permission first.

Fig22 Home Page

ML Algorithm Results Page: It displays the resultant values of the dataset when tuned with Random Forest Classifier Algorithm.



RANDOM FOREST CLASSIFIER ON THE DATASET

```
Results ====== Correctly Classified Instances 1481 73.5353 % Incorrectly Classified Instances 533 26.4647 % Kappa statistic 0.4707 K&B Relative Info Score 78830.7946 % K&B Information Score 788.2581 bits 0.3914 bits/instance Class complexity | order 0 2013.8886 bits 0.9999 bits/instance Class complexity | scheme 13144.1091 bits 6.5264 bits/instance Complexity improvement (Sf) -11130.2205 bits -5.5264 bits/instance Mean absolute error 0.3132 Root mean squared error 0.3991 Relative absolute error 62.6439 % Root relative squared error 79.8145 % Total Number of Instances 2014

=== Detailed Accuracy By Class === TP Rate FP Rate Precision Recall F-Measure ROC Area Class 0.731 0.261 0.741 0.731 0.736 0.836 1 0.739 0.269 0.73 0.739 0.735 0.836 -1 Weighted Avg. 0.735 0.265 0.735 0.735 0.735 0.836 Results For Class -1-

Precision= 0.7407776669990029

Recall= 0.7312992125984252

F-measure= 0.7360079247152056

Results For Class -2-

Precision= 0.7299703264094956

Recall= 0.7394789579158316

F-measure= 0.7346938775510202
```

Fig23 ML Algorithm Results Page

Detecting URLs Page: By providing the URL which need to be checked in the search box, we get the information of that URL whether it is Phished website or not, Spam or not, Malicious website or not, it contains malware data or not.

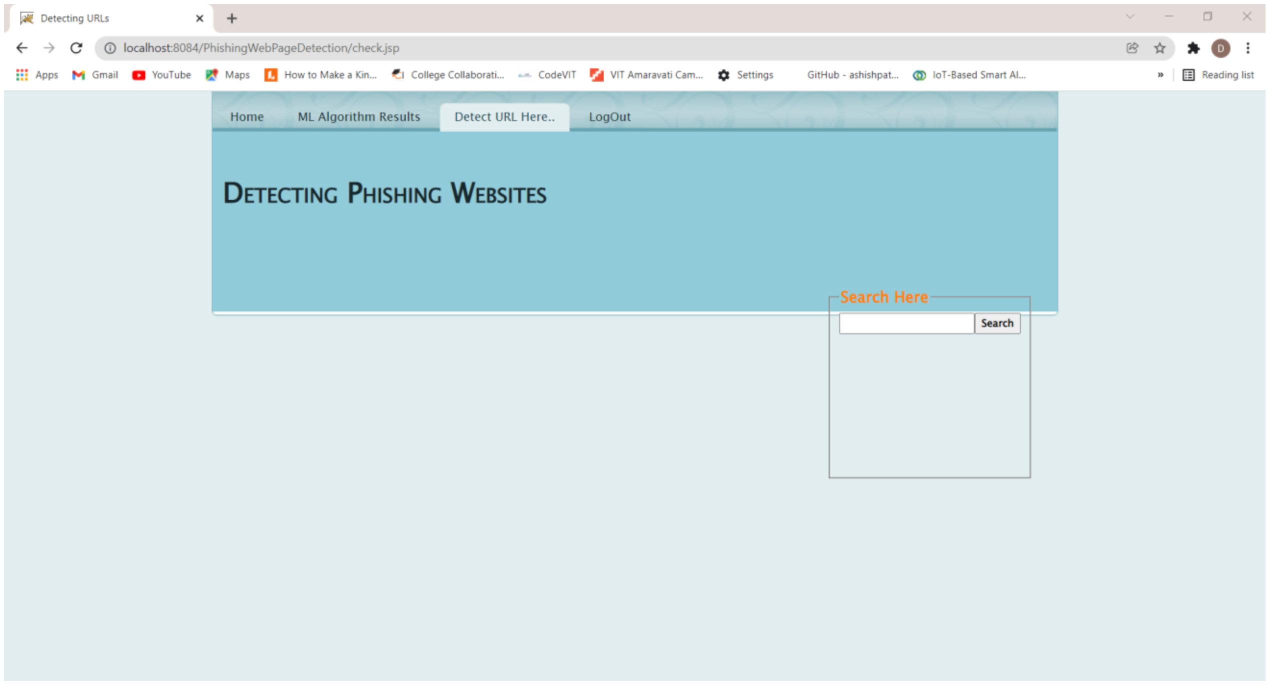
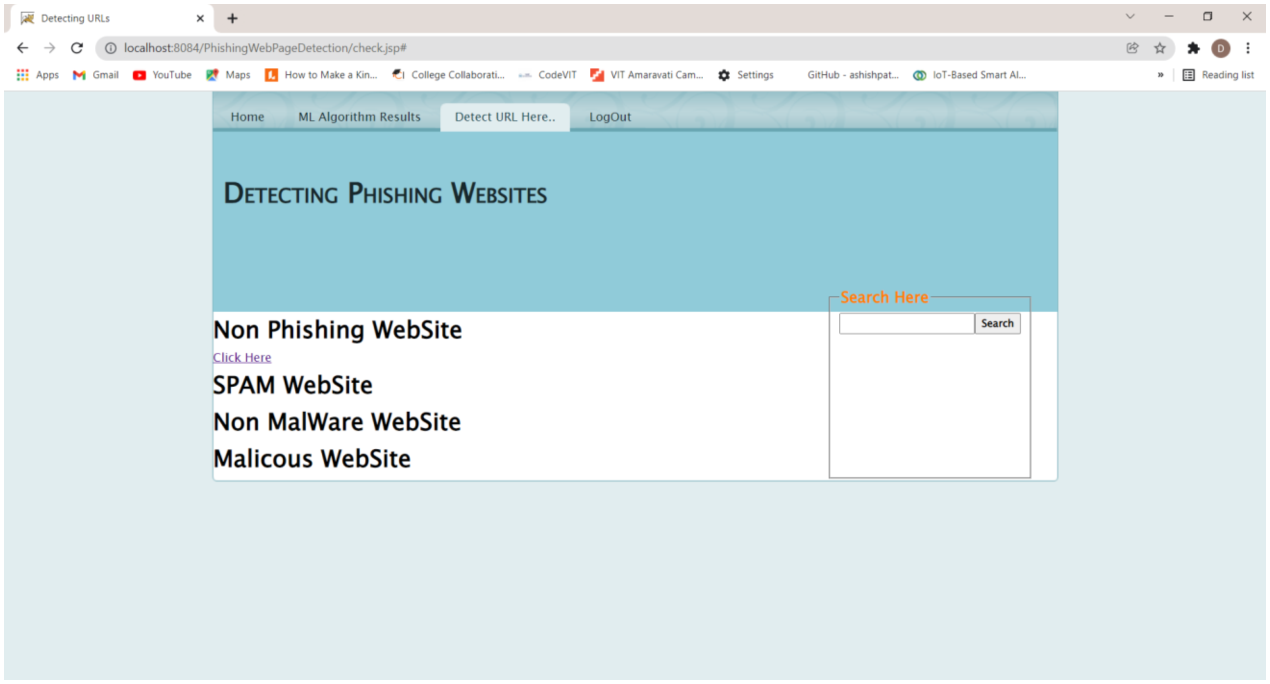


Fig24 Detecting URLs Page



DETECTING PHISHING WEBSITES

Non Phishing WebSite

Click Here

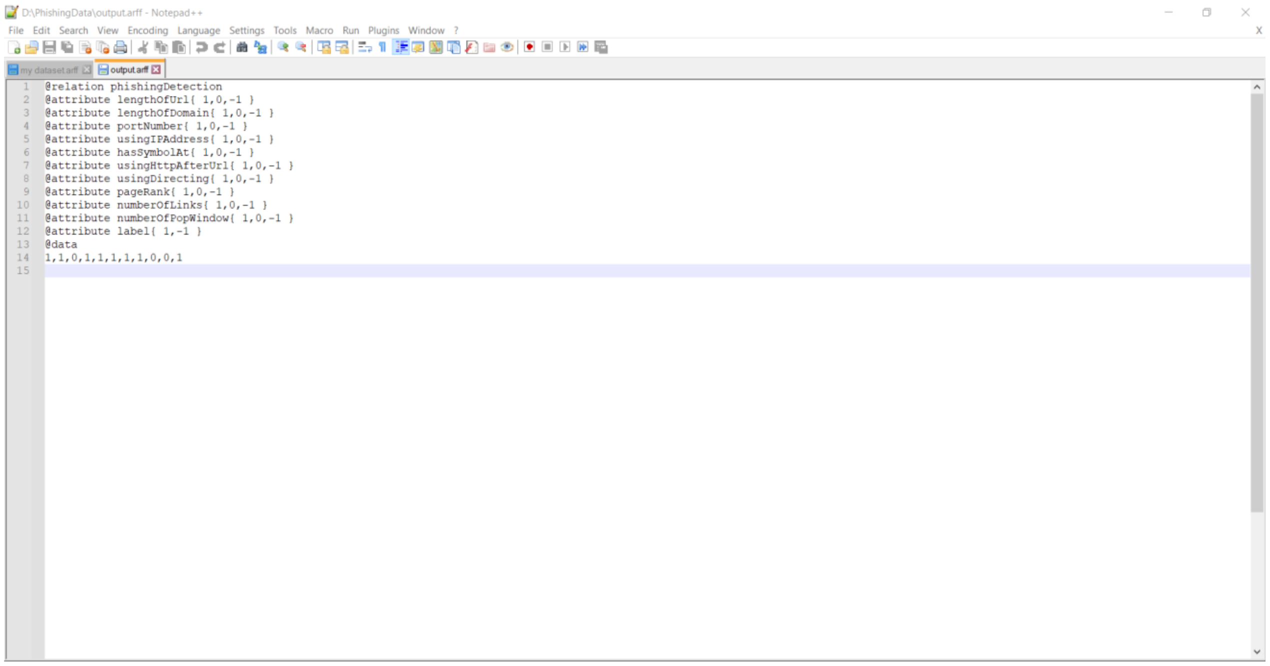
SPAM WebSite

Non MalWare WebSite

Malicous WebSite

Fig25 Results after detecting *http://google.com*

Background process of that URL detection : : If the obtained values have values>=0 then it is a Non-Phishing website.


D:\PhishingData\output.arff - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
my dataset.arff
output.arff

```
@relation phishingDetection
@attribute lengthOfUrl{ 1,0,-1 }
@attribute lengthOfDomain{ 1,0,-1 }
@attribute portNumber{ 1,0,-1 }
@attribute usingIPAddress{ 1,0,-1 }
@attribute hasSymbolAt{ 1,0,-1 }
@attribute usingHttpAfterUrl{ 1,0,-1 }
@attribute usingDirecting{ 1,0,-1 }
@attribute pageRank{ 1,0,-1 }
@attribute numberOfLinks{ 1,0,-1 }
@attribute numberOfPopWindow{ 1,0,-1 }
@attribute label{ 1,-1 }
@data
1,1,0,1,1,1,1,1,0,0,1
```

Fig26 Background results after detecting *http://google.com*

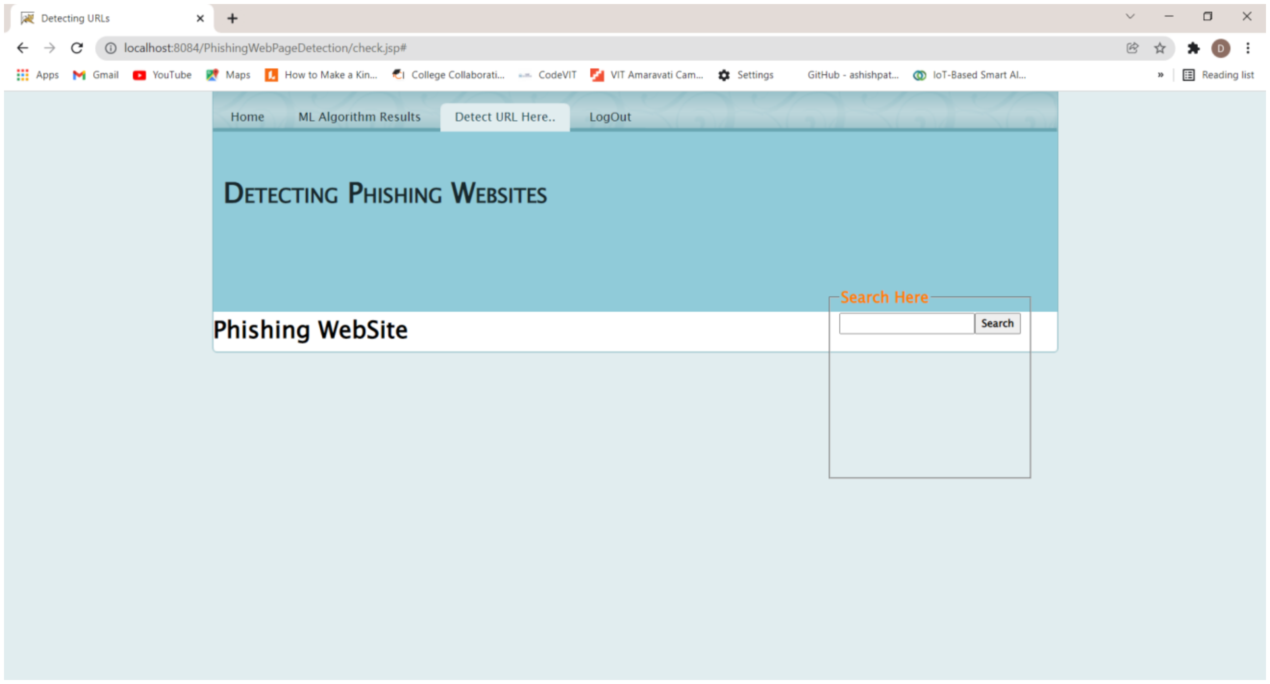
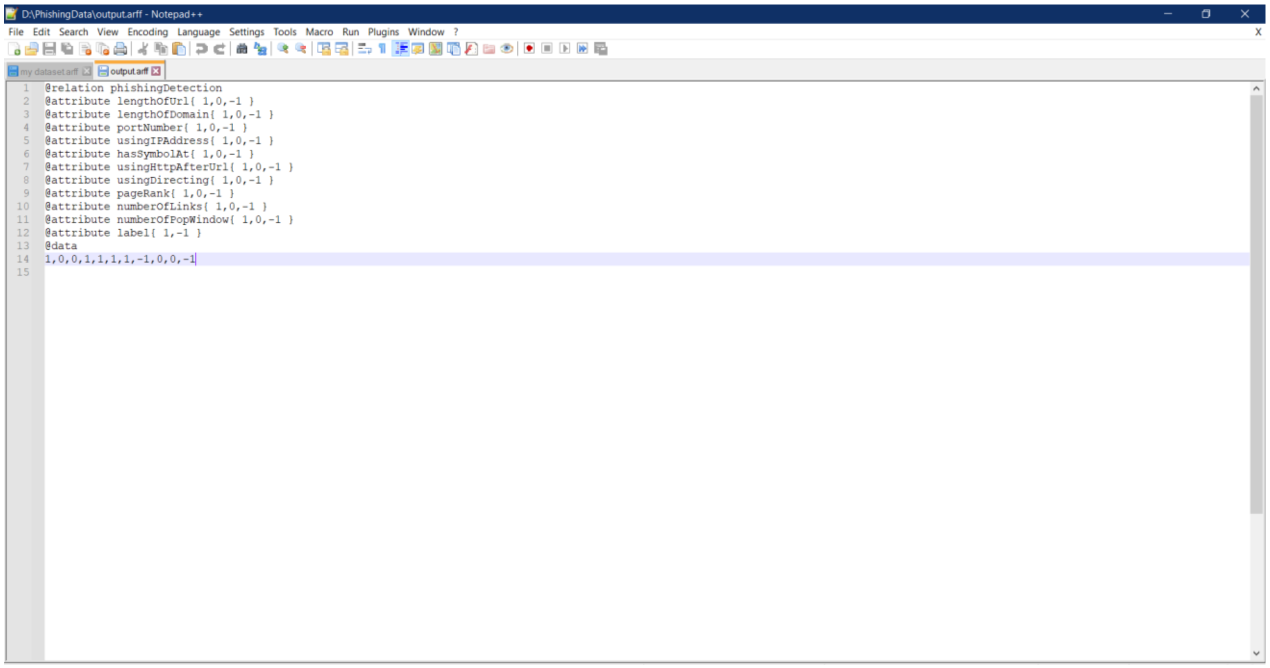


Fig27 Results after detecting *http://asesoresvelfit.com/media/datacredito.co/*

Background process of that URL detection : If the obtained values have values<0 then it is a Phishing website.


D:\PhishingData\output.arff - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
my dataset.arff
output.arff

```
@relation phishingDetection
@attribute lengthOfUrl{ 1,0,-1 }
@attribute lengthOfDomain{ 1,0,-1 }
@attribute portNumber{ 1,0,-1 }
@attribute usingIPAddress{ 1,0,-1 }
@attribute hasSymbolAt{ 1,0,-1 }
@attribute usingHttpAfterUrl{ 1,0,-1 }
@attribute usingDirecting{ 1,0,-1 }
@attribute pageRank{ 1,0,-1 }
@attribute numberOfLinks{ 1,0,-1 }
@attribute numberOfPopWindow{ 1,0,-1 }
@attribute label{ 1,-1 }
@data
1,0,0,1,1,1,1,-1,0,0,-1
```

Fig28 Background results after detecting *http://asesoresvelfit.com/media/datacredito.co/*

CHAPTER 5

CONCLUSION AND FUTURE WORK

With the obtained result, it can be concluded that the Random Forest Classifier model provides the best and highest accuracy. Whereas ensemble algorithms on the other hand have also been proved to be convenient as they are fast in speed and performance and are using more than one classifier for prediction and also gives better performance. Nowadays, website phishing is more damaging. It is becoming a big threat to people’s daily life and networking environment. In these attacks, the intruder puts on an act as if it is a trusted organization with an intention to purloin liable and essential information. The methodology we discovered is a powerful technique to detect the phished websites and can provide more effective defenses for phishing attacks of the future.

Ensemble algorithms' core principle is to combine numerous weak learners into a stronger one; this is perhaps the most important reason why ensemble-based learning is employed in practise for most classification issues. It's worth noting that there's no guarantee that combining many classifiers will always outperform the best individual classifier in an ensemble classifier. The findings stimulate future research into adding more features to the dataset, which could increase the effectiveness of these models. As a result, machine learning models could be used with other phishing detection techniques, such as List-Base methods, to achieve better results.Besides, we will explore to propose and develop a new mechanism to extract new features from the website to keep up with new techniques in phishing attacks.

Future Implementations:

- To save the information of websites which are being checked, in a database and intimidating the user with a pop-up or a notification when accessed a fraud website.
- • In a future edition of the app, there may be an option to send an SMS message straight to the blacklisted website. The list might be accessed as an attachment by the software. This text message integration feature would improve the application's usefulness.
- • In the future, we plan to make the phishing detection system a scalable web service with online learning so that new phishing assault patterns can be quickly learned and our models' accuracy can be improved with greater feature extraction.

CHAPTER 6

APPENDIX

Anaconda - Jupyter Notebook Python Code

Importing basic packages
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```
Loading the data
```
data0 = pd.read_csv('D:\PhishingData\PhishData.csv')
data0.head()
```
Checking the shape of the dataset
```
data0.shape
```
Listing the features of the dataset
```
data0.columns
```
Information about the dataset
```
data0.info()
```
Plotting the data distribution
```
data0.hist(bins = 50,figsize = (15,15))
plt.show()
```
Correlation heatmap
```
plt.figure(figsize=(15,13))
sns.heatmap(data0.corr())
plt.show()
data0.describe()
```
Dropping the Domain column
```
data = data0.drop(['id'], axis = 1).copy()
```
checking the data for null or missing values
```
data.isnull().sum()
```
shuffling the rows in the dataset so that when splitting the train and test set are equally distributed
```
data = data.sample(frac=1).reset_index(drop=True)
data.head()
```
Sepratating & assigning features and target columns to X & y
```
X = data.drop('Label',axis=1)
y = data['Label']
X.shape, y.shape
```
Splitting the dataset into train and test sets: 80-20 split
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 12)
X_train.shape, X_test.shape
```

<u>ML Algorithms training:</u>

importing packages
```
from sklearn.metrics import accuracy_score
```
Creating holders to store the model performance results
```
ML_Model = []
acc_train = []
acc_test = []
```

function to call for storing the results
```
def storeResults(model, a,b):
  ML_Model.append(model)
  acc_train.append(round(a, 3))
  acc_test.append(round(b, 3))
```

<u>Decision Tree model :</u>

```
from sklearn.tree import DecisionTreeClassifier
```
instantiate the model
```
tree = DecisionTreeClassifier(max_depth = 5)
```
fit the model
```
tree.fit(X_train, y_train)
```
predicting the target value from the model for the samples
```
y_test_tree = tree.predict(X_test)
y_train_tree = tree.predict(X_train)
```
computing the accuracy of the model performance
```
acc_train_tree = accuracy_score(y_train,y_train_tree)
acc_test_tree = accuracy_score(y_test,y_test_tree)
print("Decision Tree: Accuracy on training Data:
{:.3f}".format(acc_train_tree))
print("Decision Tree: Accuracy on test Data: {:.3f}".format(acc_test_tree))
```
checking the feature improtance in the model
```
plt.figure(figsize=(9,7))
n_features = X_train.shape[1]
plt.barh(range(n_features), tree.feature_importances_, align='center')
plt.yticks(np.arange(n_features), X_train.columns)
plt.xlabel("Feature importance")
plt.ylabel("Feature")
plt.show()
```
storing the results. The below mentioned order of parameter passing is important.
```
storeResults('Decision Tree', acc_train_tree, acc_test_tree)
```

<u>Random Forest model:</u>

```
from sklearn.ensemble import RandomForestClassifier
```
instantiate the model
```
forest = RandomForestClassifier(max_depth=5)
```
fit the model
```
forest.fit(X_train, y_train)
```
predicting the target value from the model for the samples
```
y_test_forest = forest.predict(X_test)
y_train_forest = forest.predict(X_train)
```
computing the accuracy of the model performance
```
acc_train_forest = accuracy_score(y_train,y_train_forest)
acc_test_forest = accuracy_score(y_test,y_test_forest)
print("Random forest: Accuracy on training Data:
{:.3f}".format(acc_train_forest))
print("Random forest: Accuracy on test Data: {:.3f}".format(acc_test_forest))
```
checking the feature improtance in the model
```
plt.figure(figsize=(9,7))
n_features = X_train.shape[1]
plt.barh(range(n_features), forest.feature_importances_, align='center')
plt.yticks(np.arange(n_features), X_train.columns)
plt.xlabel("Feature importance")
plt.ylabel("Feature")
plt.show()
```
storing the results. The below mentioned order of parameter passing is important.
```
storeResults('Random Forest', acc_train_forest, acc_test_forest)
```

<u>Support Vector Machines :</u>

```
from sklearn.svm import SVC
```

instantiate the model

```
svm = SVC(kernel='linear', C=1.0, random_state=12)
```

fit the model

```
svm.fit(X_train, y_train)
```

predicting the target value from the model for the samples

```
y_test_svm = svm.predict(X_test)
y_train_svm = svm.predict(X_train)
```

computing the accuracy of the model performance

```
acc_train_svm = accuracy_score(y_train,y_train_svm)
acc_test_svm = accuracy_score(y_test,y_test_svm)
print("SVM: Accuracy on training Data: {:.3f}".format(acc_train_svm))
print("SVM : Accuracy on test Data: {:.3f}".format(acc_test_svm))
```

storing the results. The below mentioned order of parameter passing is important.

```
storeResults('SVM', acc_train_svm, acc_test_svm)
```

<u>Comparison Of Models</u> :

```
creating dataframe
results = pd.DataFrame({ 'ML Model': ML_Model,
    'Train Accuracy': acc_train,
    'Test Accuracy': acc_test})
results
```

Sorting the datafram on accuracy

```
results.sort_values(by=['Test Accuracy', 'Train Accuracy'], ascending=False)
```

Java NetBeans-JSP Codes

Adminhome.jsp :

```
 function validation()
            {
                var a = document.form.userid.value;
                var b = document.form.pass.value;
                if(a=="")
                {
                    alert("Enter your UserId");
                    document.form.userid.focus();
                    return false;
                }
                if(b=="")
                {
                    alert("Enter your Password");
                    document.form.pass.focus();
                    return false;
                }
            }
        </script>
            </center>
            <%
            try{
                 BufferedReader br = null;
        int numFolds = 10;
        br = new BufferedReader(new FileReader("D:\\PhishingData\\my
dataset.arff"));
```

```
        Instances trainData = new Instances(br);
        trainData.setClassIndex(trainData.numAttributes() - 1);
        br.close();
        RandomForest rf = new RandomForest();

        rf.setNumTrees(100);

     //   rf.buildClassifier(trainData);
        Evaluation evaluation = new Evaluation(trainData);
        evaluation.crossValidateModel(rf, trainData, numFolds, new Random(1));
        System.out.println(evaluation.toSummaryString("\nResults\n======\n",
true));
        System.out.println(evaluation.toClassDetailsString());
        System.out.println("Results For Class -1- ");
        System.out.println("Precision=  " + evaluation.precision(0));
        System.out.println("Recall=  " + evaluation.recall(0));
        System.out.println("F-measure=  " + evaluation.fMeasure(0));
        System.out.println("Results For Class -2- ");
        System.out.println("Precision=  " + evaluation.precision(1));
        System.out.println("Recall=  " + evaluation.recall(1));
        System.out.println("F-measure=  " + evaluation.fMeasure(1));
        /**************************************************/
        out.println(evaluation.toSummaryString("\nResults\n======\n", true));
        out.println("<br></br>");
        out.println(evaluation.toClassDetailsString());
        out.println("Results For Class -1- ");
        out.println("<br></br>");
        out.println("Precision=  " + evaluation.precision(0));
        out.println("<br></br>");
        out.println("Recall=  " + evaluation.recall(0));
        out.println("<br></br>");
        out.println("F-measure=  " + evaluation.fMeasure(0));
        out.println("<br></br>");
        out.println("Results For Class -2- ");
        out.println("<br></br>");
        out.println("Precision=  " + evaluation.precision(1));
        out.println("<br></br>");
        out.println("Recall=  " + evaluation.recall(1));
        out.println("<br></br>");
        out.println("F-measure=  " + evaluation.fMeasure(1));
        out.println("<br></br>");
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
            %>
</body>
</html>
```

check.jsp:

```
function validation()
            {
                var a = document.form.userid.value;
                var b = document.form.pass.value;
                if(a=="")
                {
                    alert("Enter your UserId");
                    document.form.userid.focus();
                    return false;
                }
                if(b=="")
                {
                    alert("Enter your Password");
                    document.form.pass.focus();
                    return false;
                }
            }
        </script>
          <div style="position:absolute; right:30px; top:230px;">
                <fieldset>
                   <legend><font color="#FF8000"><strong><font size="4">Search
Here</font></strong></font></legend>
                    <table height="160">
                        <form action="#" method="post" name="form"
onsubmit="return validation();">
                            <input type="text" name="urlname" value="">
</input>
                                <input type="submit" name="submit"
value="Search"></input>
                      </form>
                    </table>
                </fieldset>
                <br><br><br>
           </div>
```

```
            </center>
            <%
          String name=request.getParameter("urlname");
            try{
                //JOptionPane.showMessageDialog(null, "hello");
                int cntt=0;
                name=request.getParameter("urlname");
                //JOptionPane.showMessageDialog(null, "hello"+name);
                FileWriter fw=new FileWriter("D:\\PhishingData\\cst.txt");
                BufferedWriter bw=new BufferedWriter(fw);
                bw.write(name);
                bw.close();
                fw.close();
                Features ob=new Features();
                ob.main("D:\\PhishingData\\cst.txt");
                /*******************************/
                FileReader fr1=new
FileReader("D:\\PhishingData\\output.arff");
                BufferedReader br1=new BufferedReader(fr1);
                // String data=br.readLine();
                String data1=br1.readLine();
                int count1=0;
                while(data1!=null)
                {
                //JOptionPane.showMessageDialog(null, "hello"+name);
                if(data1.contains("@"))
                data1=br1.readLine();
                 //bw.write(data1+"\n");
                if(data1.contains("-1"))
                {
                   // out.println("<h1>Phishing WebSite</h1>");
                    //JOptionPane.showMessageDialog(null,"Phishing WebSite");
                    count1=0;
                }
                else if(data1.contains("1"))
                {
                     //out.println("<h1>Non Phishing WebSite</h1>");
                     //JOptionPane.showMessageDialog(null,"Non Phishing
WebSite");
                    count1=1;
                }
         //   }
           data1=br1.readLine();
        }
                System.out.println("Count1::"+count1);
                if(count1>0)
                {
                    out.println("<h1>Non Phishing WebSite</h1>");
                   // int spamcnt=new Spam_feature().main(name);
                   %>
                   <a href="<%=name%>">Click Here</a>
                   <%
                }
                else
                {
                    out.println("<h1>Phishing WebSite</h1>");
                }
                /*************SPAM DETECTION***************/
                int spmcount=0;
                URL url = new URL(name);
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream in = urlConnection.getInputStream();
byte[] data = new byte[8192];
int length;

while ((length = in.read(data)) != -1) {
    System.out.print(new String(data, 0, length));
    String[] d=new String(data, 0, length).split("htttp://");
    for(int i=0;i<d.length;i++)
    {
    if(d[i].contains("year")||d[i].contains("event | product
name")||d[i].contains("adult")||d[i].contains("porn")||d[i].contains("cheap")|
|d[i].contains("free")||d[i].contains("offer"))
    {
    spmcount++;
    cntt++;
    }
        }
}
in.close();
urlConnection.disconnect();
             if(spmcount>0)
                 out.println("<h1>SPAM WebSite</h1>");
             else
                 out.println("<h1>Non SPAM WebSite</h1>");
             /****************MALWARE DETECTION*******************/
             FileReader frr=new FileReader("D://malware_data.txt");
             BufferedReader brr=new BufferedReader(frr);
             String mdata=brr.readLine();
             int malcnt=0;
             while(mdata!=null)
             {
             if(mdata.equalsIgnoreCase(name)){
                 malcnt++;
                 cntt++;
             }
             mdata=brr.readLine();
             }
             if(malcnt>0)
                 out.println("<h1>MaleWare WebSite</h1>");
```

```
            else
                out.println("<h1>Non MalWare WebSite</h1>");
                /******Malicious**************************/
              if(malcnt>0||spmcount>0||count1>0)
              out.println("<h1>Malicous WebSite</h1>");
            else
                out.println("<h1>Non Malicious WebSite</h1>");
           }
           catch(Exception e)
           {
               e.printStackTrace();
           }
           %>
</body>
</html>
```

Userlogin.jsp:

```
<body>
        <h1>Hello World!</h1>
        <%
        try{
            String uid,pwd;
            uid=request.getParameter("userid");
            pwd=request.getParameter("pass");

if(uid.equalsIgnoreCase("arshitha")&&pwd.equalsIgnoreCase("arshi"))
            {
                response.sendRedirect("home.jsp");
            }
            else
            {
               response.sendRedirect("index.html?message=invlid credential");
            }
        }
        catch(Exception e)
        {
        e.printStackTrace();
        }
        %>
    </body>
```

Hereby, I am attaching link to access the files of the project :
https://drive.google.com/drive/folders/1VV0pkdzax9nT71VdbEv0U-aKaGOOWqqj?usp=sharing

REFERENCES

[1] https://www.ijert.org/detection-of-phishing-websites-using-machine-learning

[2] https://www.researchgate.net/publication/354308182_Phishing_Website_Detection_Using_ML

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8504731/

[4] https://www.activestate.com/blog/phishing-url-detection-with-python-and-ml/

[5] Wu, Jiajing, et al. 2020 Who are the phishers? phishing scam detection on ethereum via network embedding. IEEE Transactions on Systems, Man, and Cybernetics: Systems

[6] Niu, Xiaofei, Guangchi Liu, and Qing Yang 2020 OpinionRank: Trustworthy Website Detection using Three Valued Subjective Logic. IEEE Transactions on Big Data

[7] Crane, C. (2019). 20 Phishing Statistics to Keep You from Getting Hooked in 2019 - Hashed Out by The SSL StoreTM. Hashed Out by The SSL StoreTM, Jul. 24, 2019. https://www.thesslstore.com/blog/20-phishing-statistics-to-keep-you-from-getting-hooked-in-2019/, accessed on Mar. 25, 2020.

[8] J. Shad and S. Sharma, A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology, pp. 425430, 2018.

[9] S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, A New Method for Detection of Phishing Websites: URL Detection, in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no. Icicct, pp. 949952.

[10] A. Desai, J. Jatakia, R. Naik, and N. Raul, Malicious web content detection using machine leaning, RTEICT 2017 – 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc., vol. 2018Janua, pp. 14321436, 2018.

[11] Salihovic, I., Serdarevic, H., Kevric, J. (2018). The role of feature selection in machine learning for detection of spam and phishing attacks. Advanced Technologies, Systems, and Applications III, 476-483. https://doi.org/10.1007/978-3-030-02577-9_47

[12]Rishikesh Mahajan (2018) “Phishing Website Detection using Machine Learning Algorithms”

[13] Purvi Pujara, M. B.Chaudhari (2018) “Phishing Website Detection using Machine Learning : A Review”

[14] Satish.S, Suresh Babu.K (2013) “Phishing Websites Detection Based On Web Source Code And Url In The Webpage”

[15] V. B. et al, “study on phishing attacks,” International Journal of Computer Applications, 2018

[16] R. C. Dodge Jr, C. Carver, and A. J. Ferguson, “Phishing for user security awareness,” computers & security, vol. 26, no. 1, pp. 73–80, 2007

[17]https://www.researchgate.net/publication/337049054_Phishing_Websites_Detection_Using_Machine_Learning

[18] J. Shad and S. Sharma, “A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology,” pp. 425–430, 2018.

[19] S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, “A New Method for Detection of Phishing Websites: URL (ICICCT), 2018, vol. 0, no. Icicct, pp. 949–952.

[20] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. URL-based heuristic,” 2014 Int. Conf. Comput. Manag. Telecommun. ComManTel 2014, pp. 298–303, 2014.

[21] A. A. Ahmed and N. A. Abdullah, “Real time detection of Mob. Commun. Conf. IEEE IEMCON 2016, 2016.