A MINOR PROJECT REPORT

ON

"DETECTION OF PHISHING WEBSITES USING MACHINE LEARNING"

In Partial Fulfilment of the Requirement for the

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

1. Rohit Kundan Sonawane	TE-B-59
2. Sakshi Pandurang Mahajan	TE-A-55
3. Vaishnavi Vilas Mahajan	TE-A-29
4. Chetan Jayram Mahajan	TE-A-52

UNDER THE GUIDANCE OF
Prof. Mr. Ashish T Bhole Sir

DEPARTMENT OF COMPUTER ENGINEERING

SSBT's College of Engineering and Technology, Bambhori Jalgaon - 425001 [MS]

2021-2022

SSBT’s College of Engineering and Technology, Bambhori
Jalgaon - 425001 [MS]

Department of Computer Engineering

CERTIFICATE

This is certify that the project entitled

“DECTION OF PHISHING WEBSITE USING MACHINE LEARNING“

submitted by

1. Rohit Kundan Sonawane	TE-B-59
2. Sakshi Pandurang Mahajan	TE-A-55
3. Vaishnavi Vilas Mahajan	TE-A-29
4. Chetan Jayram Mahajan	TE-A-52

in partial fulfillment of the degree of Bachelor of Engineering in Computer Engineering has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: / /

Place : Jalgaon

(Prof. A.T.BHOLE)
Project Guide

(Prof. Manoj Patil Sir)
HOD, Computer Department

(Dr. G. K. Patnaik Sir)
Principal

Acknowledgements

We would like to express our deep gratitude and sincere thanks to all who helped us in completing this project report successfully. Many thanks to almighty God who gave us the strength to do this.

Our sincere thanks to Principal Prof. Dr. Girish. K. Patnaik for providing the facilities to complete this Project report. We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to Prof. Dr. Krishnakant P. Adhiya, Head of the Department, whose help, stimulating suggestions and encouragement, helped us in writing this report.

We would also like to thank (Mr. Ashish Bhole), Project Guide, who has given his/her full effort in guiding us and achieving the goal as well as his encouragement to maintain the progress in track. I am also sincerely thankful to Mr. Akash D. Waghmare, Incharge of Project, for his valuable suggestions and guidance.

We would also like to appreciate the guidance given by other supervisor that has improved our presentation skills by their comments and tips. Last but not the least, we are extremely thankful to our parents and friends without whom it could not reach its successful completion.

Rohit Kundan Sonawane

Sakshi Pandurang Mahajan

Vaishnavi Vilas Gadhe

Chetan Jayram Mahajan

Contents

List of Figures

ABSTRACT

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting internet users. Phishers use spoofed e-mail, phishing software to steal personal information and financial account details such as usernames and passwords. The malicious links within the body of the message are designed to make it appear that they go to the spoofed organization using that organization's logos and other legitimate contents.This paper deals with methods for detecting phishing web sites by Natural Language Processing and by Machine learning techniques. Machine learning is a powerful tool used to strive against phishing attacks. We discuss the methods used for detection of phishing websites based on NLTK libraries such as word segmentation, stemming and lemmatization(methods of trimming words down to their roots), and tokenization(for breaking phrases, sentences and paragraph into tokens that help computer to better understand text) The fine-tuned parameters are useful in selecting the apt machine learning algorithm for classifying the phishing sites and benign sites. We've classified them into 'Good' sites and 'Bad' sites.

Chapter 1

Introduction

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting internet users. Phishers use spoofed e-mail, phishing software to steal personal information and financial account details such as usernames and passwords.. This paper deals with methods for detecting phishing web sites by doing feature extraction of urls by Machine learning techniques and Natural Language Processing. The organization of this Chapter is as follows. Section 1.1 describes Background of the project. Motivation of the project is represented in Section 1.2 Section 1.3 represents Problem statement of the project. Scope of the project is described in Section 1.4. Section 1.5 describes Objective of the project. Section 1.6 describes the selection of life cycle model. Section 1.7 shows the organization of report and finally, the Summary is described in 1.8 Section

1.1 Background

In recent years, advancements in Internet and cloud technologies have led to a significant increase in electronic trading in which consumers make online purchases and transactions. This growth leads to unauthorized access to users' sensitive information and damages the resources of an enterprise. Phishing content and gain their information. In terms of website interface and uniform resource locator (URL), most phishing webpages look identical to the actual webpages. Various strategies for detecting phishing websites, such as blacklist, heuristic, Etc., have been suggested. However, due to inefficient security technologies, there is an exponential increase in the number of victims. The anonymous and uncontrollable framework of the Internet is more vulnerable to phishing attacks. Existing research works show that the performance of the phishing detection system is limited. There is a demand for an intelligent technique to protect users from the cyber-attacks. In this study, the author proposed a URL detection technique based on machine learning approaches.

.

1.2 Motivation

As we discussed earlier that phishing is a major threat to our society these days and evolution of AI enabled us so many ways to deal with such kind of scams. So with the help of machine learning algorithms we can build softwares that can detect phishing websites. This remain a motivation for us to build a model that can detect phishing websites and tell users about them.

1.3 Problem Defination

URLs sometimes known as “Web links” are the primary means by which users locate information in the Internet. Aim of the phishers is to acquire critical information like username, password and bank account details. Our aim is to derive classification models that detect phishing urls using machine learning and natural language processing. In JuPyter environment.

A general statement of the phishing website detection using machine learning can be formulated as follows

Given a website into the input, detect whether it is phishing or not.

1.4 Scope

• Around 83 percent of IT teams in Indian organizations said the number of phishing emails targeting their employees increased during 2020, according to the findings of a global survey titled ‘Phishing Insights 2021’ by Sophos, a cyber security company.
• The good news is that most organizations in India (98) have implemented cybersecurity awareness programme to combat phishing. Respondents said

they use computer-based training programme (67), human-led training programme (60), and phishing simulations (51).

- So taking these numbers and current scenario in consideration, there is huge scope for phishing website detection with the help of latest technologies such as AI, Machine learning

1.5 Objective

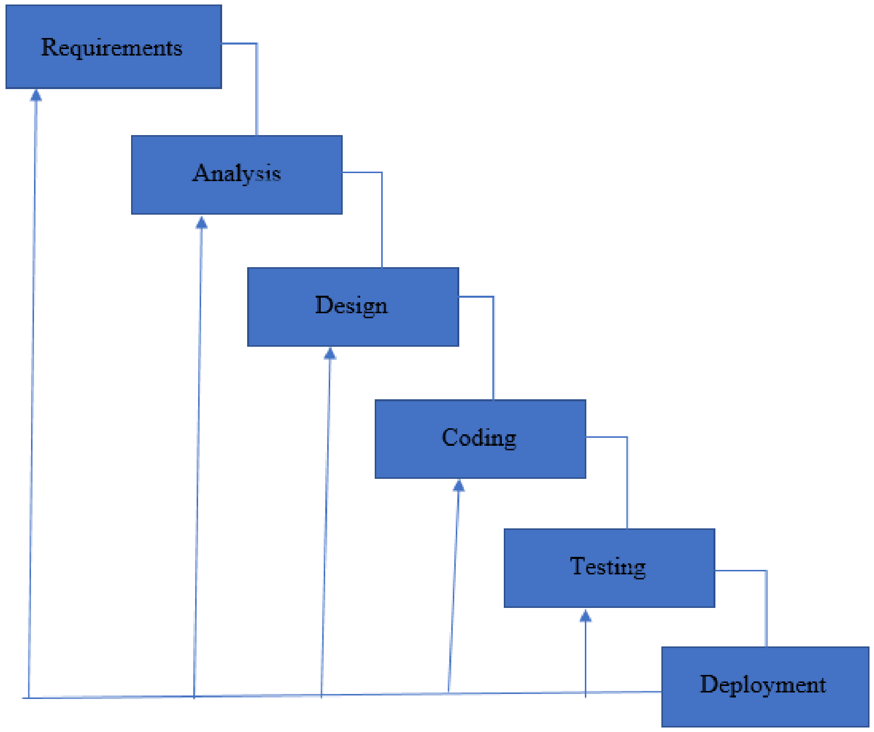
- The objective of this thesis will be
- To explain what phishing websites are? And how they are major threat to peoples.
- To collect phishing websites database and perform processing on them
- After by applying various feature extraction techniques, fitting of the model with machine learning algorithm
- Improving accuracy of the model
- Deployment of model of web-page and make it ready mo use for end users.
- Users can enter their website on our website and check whether it is Phishing or not.

1.6 Selection of Life Cycle Model for Development

The software development life cycle model selected for this project is the Waterfall Model. Waterfall approach was the first SDLC Model to be widely used in software engineering to ensure success of project.

It was developed by Winston W. Royce in 1970. Classical waterfall model divides the life cycle into a set of phases.

This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Below is the diagram of Waterfall model.



1.7 Organization of Report

The report is described in following way.

Chapter 1 : Titled as Introduction describes the Background, Motivation, Problem Defination, Scope and Objectives.

Chapter 2 : Title as Project Planning and Management, which presents Proposed system and Feasibility studies and represents Risk Analysis, Project. Scheduling and Cost Estimation.

Chapter 3 : Titled at System Analysis, which presents Requirement Collection and Identification, Software/Hardware Requirements, Functional and Non-Functiotal requirements and Software Requirement Specification.

Chapter 4 : Titled as System Design presents System Architecture data flow diagrams and the UML Diagrams

Chapter 5 : Titled as Conclusion concludes the minor project.

1.8 Summary

In this chapter Introduction, Problem Definition, Scope, Objective and Organization Of the Report are presented. In next chapter, The Project Planning and Management is presented.

Chapter 2

Project Planning and Managment

Project planning is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system.

The organization of chapter is as follows. Section 2.1 Feasibility Study. Risk Analysis are presented in Section 2.2. Section 2.3 presents Project Scheduling. Effort Allocation are presented in Section 2.4. Section 2.5 presents Cost Estimation. Finally, Summary is presented in Last Section 2.6.

2.1 Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

2.1.1 Technical Feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). It is an evaluation of the hardware and software and how it meets the need of the proposed system. This website is developed using Html, Css, Flask, etc. Also all the other technologies used are capable of building such a platform and serve as well as maintain it for longer period of time. All the required hardware and software are easily available in the market. Hence the portal is technically feasible.

2.1.2 Economic Feasibility

Describes how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus. As the necessary hardware and the software are easily available in the market at low cost, the initial investment is the only cost incurred and does not need further enhancement. Hence it is economically feasible.

2.1.3 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

2.2 Risk Analysis

Risk Analysis and Management is a key project management practice to ensure that the least number of surprises occur while your project is underway. While we can never predict the future with certainty, we can apply a simple and streamlined risk management process to predict the uncertainties in the projects and minimize the occurrence or impact of these uncertainties. This improves the chance of successful project completion and reduces the consequences of those risks.

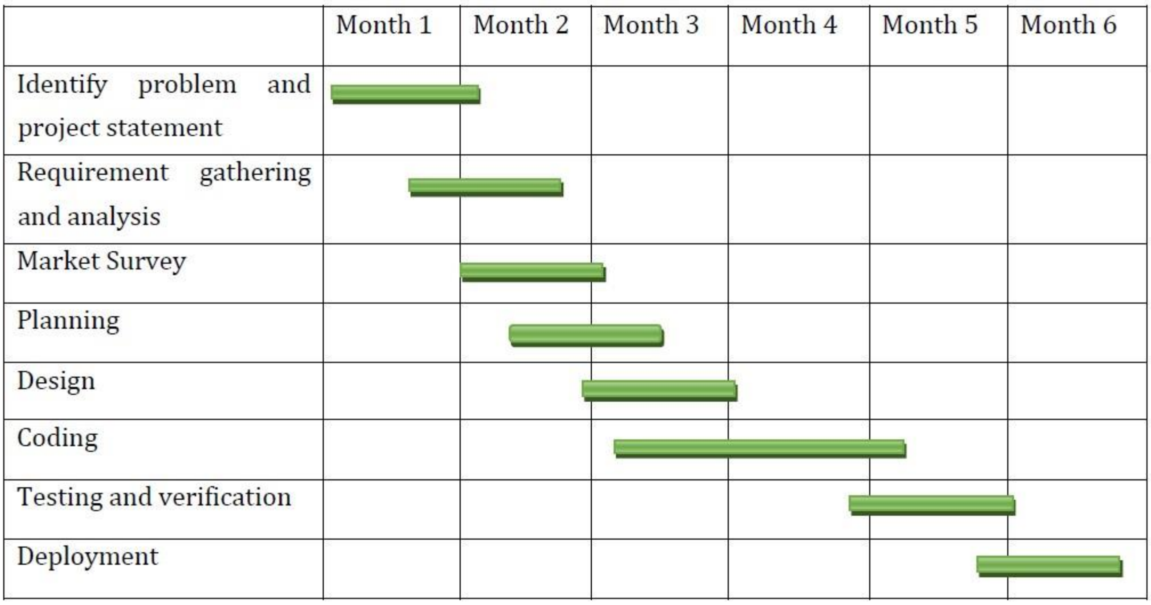
2.2.1 Risk Based on Dependencies

The system uses python 3.9 for development. Programming languages changes according current needs of the developers and for the future aspects of technology. Python will also receive time to time updates in future too. Various modules of python may change functionally after update. This issue can be avoided by creating a virtual environment.

Virtual Environment can be created in Pycharm IDE easily.

2.2.2 Risk Based on Development Teams

In today’s scenario technology rapidly changes it may be possible that the project team does not have required amount of knowledge regarding the project and the required technical proficiency to develop the system. This risk can be overcome by giving required training to developers. Bootcamps, seminars and doubt solving sessions can be conducted by the project lead to give required knowledge to the team. This will resolve the risk based on development teams.



	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
Identify problem and project statement	🟩	🟩				
Requirement gathering and analysis	🟩	🟩				
Market Survey		🟩	🟩			
Planning		🟩	🟩			
Design		🟩	🟩	🟩		
Coding			🟩	🟩	🟩	
Testing and verification				🟩	🟩	🟩
Deployment					🟩	🟩

2.3 Project Scheduling

Generally, project scheduling can be stated as the estimated time required for any project from its time to beginning to the end of the project. In details, for every task, there is a deadline because all the tasks for completion of project are planned earlier. So that, each task is scheduled to certain time limit.

In short, in project management, listing of project milestones, activities and all from starting ending date, are considered in the project scheduling. A schedule is generally used in the project planning and management of the project with kind of attributes as budget, task allocation and duration, resource allocation and all. Following figure describes the Project Scheduling

Figure 2.1: Gantt Chart for Project Schedulingl

2.4 Effort Allocation

Effort Allocation is necessary so every team member can give its best to the project. Project was divided into smaller module and task form, for simplification and easy understanding of project overall. Some modules include every team associate , presence to take advantage of team decision taking skills, and some task include some individual member to work on it with precision. We divided the project into 6 modules.

1. Collection of dataset and gathering of information
2. Planning/Requirement Analysis
3. Study of technology used
4. Selection of Life cycle Model
5. Planning and Management
6. Analysis and Design UML.

Sr. No	Work	Team associates			
		Rohit	Chetan	Vaishnavi	Sakshi
1	Collection of dataset	Yes		Yes	
2	Planning/requirement analysis	Yes	Yes		Yes
3	Study of technologies used	Yes	Yes	Yes	Yes
4	Selection of Life cycle model		Yes		Yes
5	Planning and management	Yes		Yes	
6	Analysis & UML	Yes		Yes	

2.5 Cost Estimation

Cost Estimation is an important phase for any project. It predicts if the project investment is adequate or there will shortage of capital. It presents the total cost required for development of project. Cost Estimation should be done before initiating the development to prevent loss of efforts and project failure during development. The cost estimation model i.e Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.

- Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

The cost of any software project is calculated by the formula,

C = aLb

C = cost of project, a = 1.4 (constant), b = 0.93 (constant), L = size of code

For our project considering the number of lines of code to be 21000 based on the average number of lines of code for similar projects, we can calculate the cost as follows:

Cost of project = 1.4*(3000)*0.93 = Rs3,906

2.6 Summary

This chapter describes the project planning such as its Feasibility study and also an overview of proposed system. The next chapter describes the analysis of project.

Chapter 3

Analysis

The development of computer-based information system includes the system analysis phase which produces or enhances the data model which itself is to creating or enhancing a database. There are a number of different approaches to system analysis. The analysis is the process which is used to analyze, refine and scrutinize the gathered information of entities in order to make consistence and unambiguous information. Analysis activity provides a graphical view of the entire System. System Analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will show overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer system analysis. The organization of this Chapter is as follows.

Section 3.1 represents Requirement Collection and Identification. Software Requirement and Specification are described in the Section 3.2. Section 3.3 describes summary of the chapter.

3.1 Requirement Collection and Identification

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how. In practical application, most projects will involve some combination of these various methods in order to collect a full set of useful requirements. Requirements collection is initiated when the project need is first identified and the project “solution” is to be proposed. Requirements refinement continues after the project is “selected” and as the scope is defined, aligned and approved.

Our model will only require a website which has to be entered by user. So that model can predict the output.

3.2 Software Requirements and Specifications

Software Specification will provide a broad understanding of the requirement specification of this system. Also, understand features of this system along with the requirements. Software Requirement Specification documents guide the developers in the development process and it will help to reduce the ambiguity of the requirements provided by the end-user. It’s used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

3.2.1 Model Features

The product features are high level attributes of a software or product such as software performance, user-friendly interface, security portability, etc. These attributes are defined according to the product, in this case, a software product. They are as follows:

- Model will be able to achieve 97% accuracy on prediction.
- It will be able to predict whether the website is Good or Bad.
- Output will be displayed in less than 1 sec.
- It will also provide some useful information about phishing scams and how to be safe from them.

3.2.2 Operating Environment

The software will operate within the following environment:

Hardware Requirements:-

- 2GB RAM (minimum)
- 100GB HDD (minimum)
- Intel 1.66 GHz Processor Pentium 4 (minimum)
- Internet Connectivity

Software Requirement Specification

- WINDOWS 7 or higher
- Python 3.6.0 or higher
- Visual Studio Code
- Flask
- HTML

3.2.3 User Interface

The platform’s user interface has been designed to cater the users with simplicity and to be able to perform task with minimal efforts. The home screen offers a simple input box in which user have to enter their website and click the predict button.

3.2.4 Summary

In this chapter, analysis of project is described. In the next chapter, detailed design of project is explained.

Chapter 4

Design

Design is the activity to design and model the various component of software system. The system design provides the understanding and procedural details necessary for implementing the system. Design is helpful for a better understanding of the project. It contains the UML diagrams, data flow diagrams. UML is a modeling language which is used to document the object-oriented analysis and design. The organization of this Chapter is as follows. Section 4.1 describes the work- flow of model building of the project. DFD of the project are represented in Section 4.2. Section 4.3 represents UML Diagrams (use case diagram, work-flow diagram, bar graphs, confusion matrix of results, etc. Finally, the Summary is described in last Section 4.4.

4.1 Work – Flow and Feature Extraction

- Firstly we have checked whether there is any null value present in the dataset or not and if so removed it.
- Use of RegexpTokenizer : with the help of tokenize.regexp() module, we are able to extract the tokens from strings by using regular expression with RegexpTokenizer() method.

• Use of PorterStemmer : stemmer is used to produce morphological variants of a root/base word. Stemming is desirable as it may reduce redundancy as most of the time the word stem and their derived words mean the same. And then after stemming joining of words is done.

• Feature Extraction using TfidfVectorizer: It's the main part of natural language processing . TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

• Splitting the data: Train-test-split is used to split the data into training dataset and testing dataset.

• Algorithms used: Support Vector Machine (SVM) is used. The SVM performs classification by finding the hyper plane that maximizes the margin between two classes. The vectors that define the hyper plane are the support vectors. Below is the figure showing how it works?

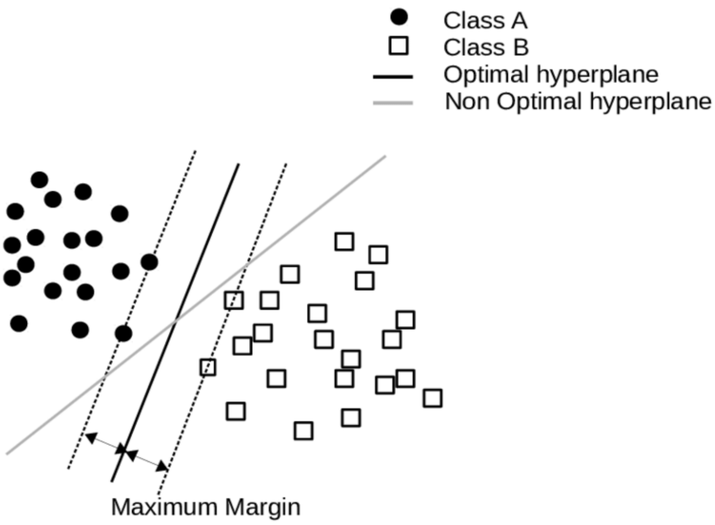


Figure 4.1: Work – flow and feature extraction

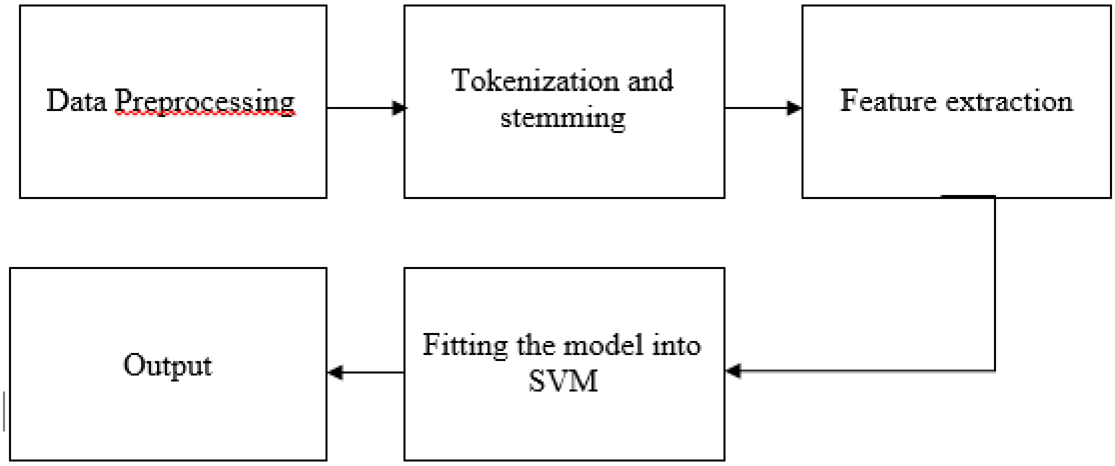
• Dumping of model using pickle: Python pickle module is used for serialising and de-serialising a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serialises” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

• Deploy model using Flask on web page: This is the last step of project. After deployment, other end users can make use of it.

4.2 UML Diagrams

4.2.1 System Architecture

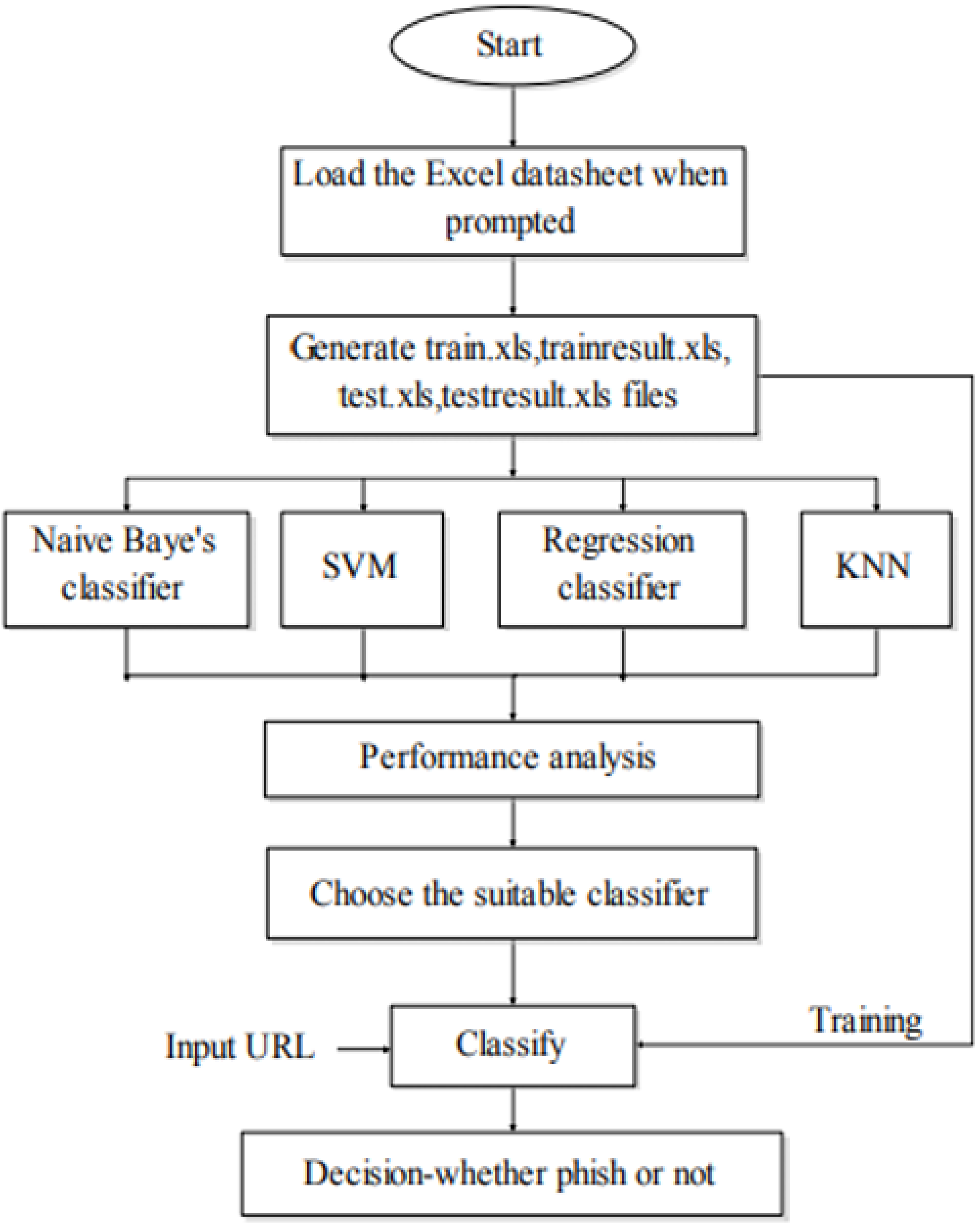
An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. it can consist of system components and the subsystems developed, that will work together to implement the overall system. One can think of system architecture as a set of representations of an existing system. System Architecture of the Phishing url detection system is shown in below figure.



4.2.2 Data Flow Diagram

A Data Flow Diagram is a structured analysis and design tool that can be used for flow- charting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFDs are often referred to as logical data ow diagrams. Data flow diagram (DFD), also called as Bubble chart is a graphical technique, which is used to represent information ow, and transformers those are applied when data moves from input to output.

Below is the Data flow diagram.


Start
Load the Excel datasheet when prompted
Generate train.xls,trainresult.xls, test.xls,testresult.xls files
Naive Baye's classifier
SVM
Regression classifier
KNN
Performance analysis
Choose the suitable classifier
Input URL
Classify
Training
Decision-whether phish or not

4.2.3 Use Case Diagram

Use case diagrams consists of actors , use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

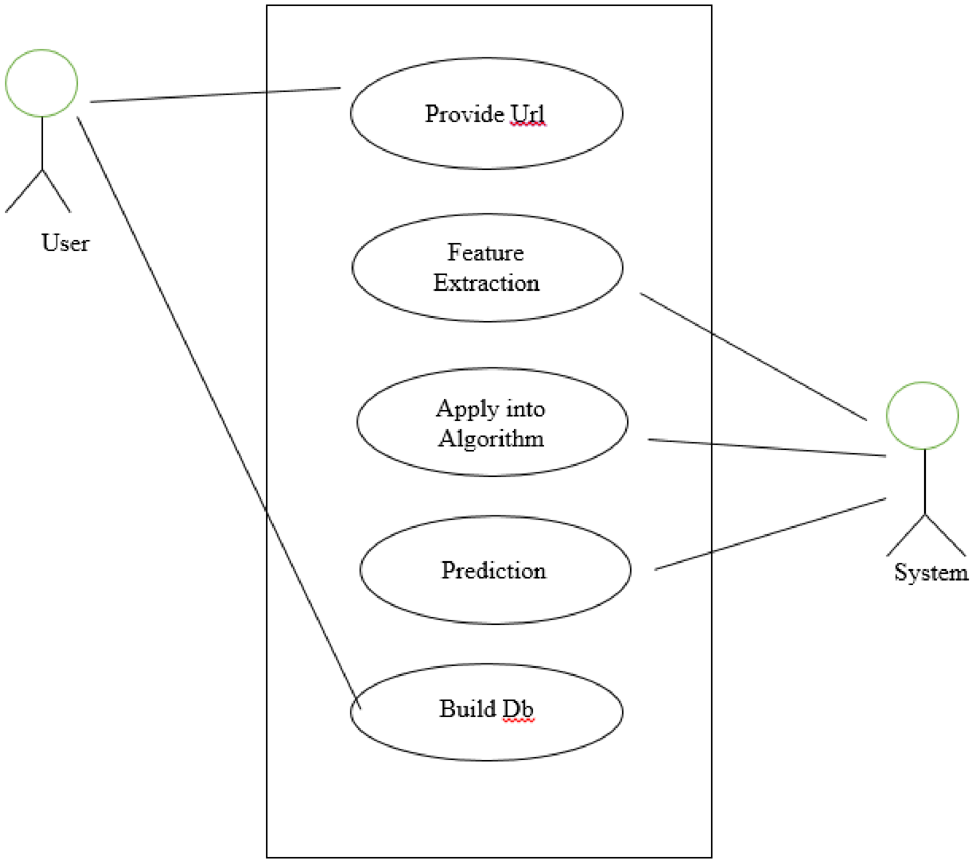


Figure 4.2.3   Use Case Diagram

4.2.4 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

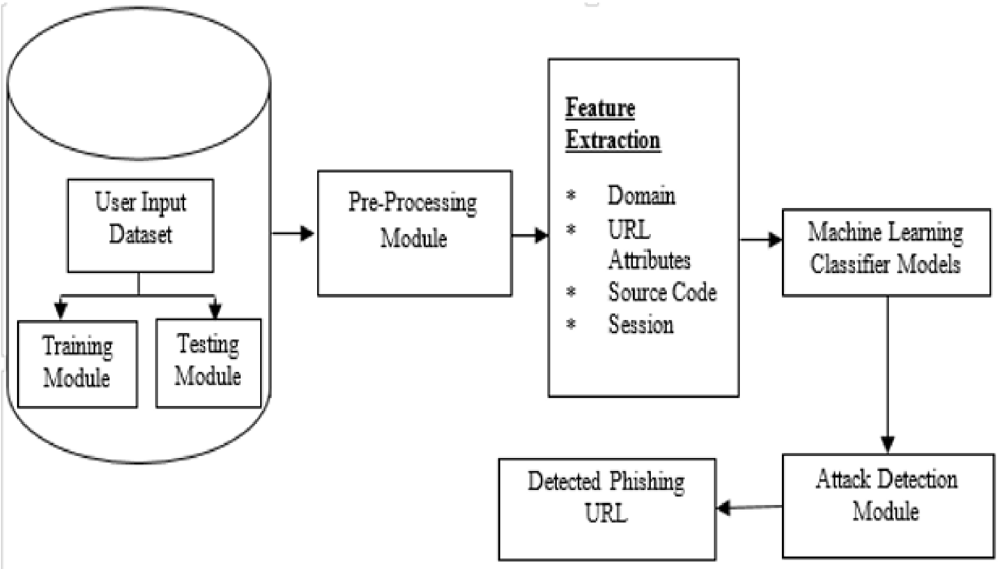


Figure 4.2.4 Activity Diagram

4.2.5 Class Diagram

The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with objectoriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints.

The class diagram is a central modeling technique that runs through nearly all objectoriented methods. This diagram describes the types of objects in the system and various kinds of static relationships which exist between them. Diagram also defines what functions will be in the classes and what kind of relationship will be among the classes.

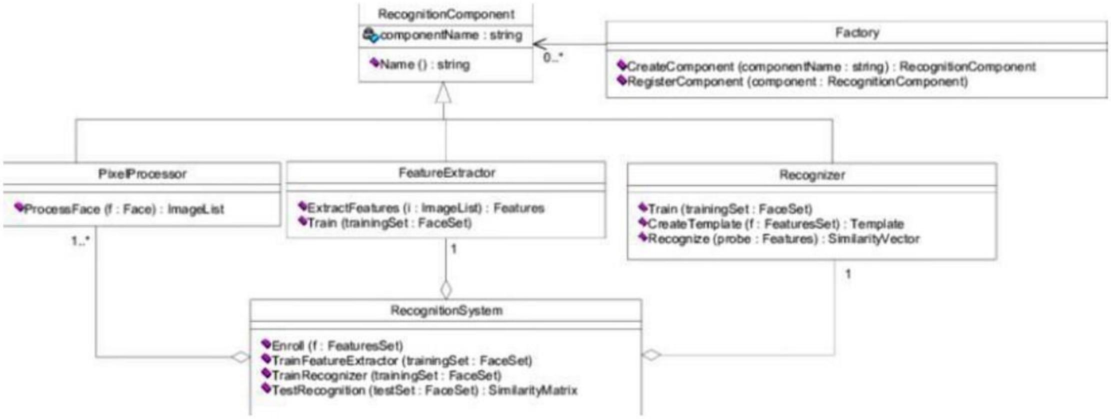

RecognitionComponent
componentName : string
Name () : string
0..*
Factory
CreateComponent (componentName : string) : RecognitionComponent
RegisterComponent (component : RecognitionComponent)
PixelProcessor
ProcessFace (f : Face) : ImageList
1..*
FeatureExtractor
ExtractFeatures (i : ImageList) : Features
Train (trainingSet : FaceSet)
1
Recognizer
Train (trainingSet : FaceSet)
CreateTemplate (f : FeaturesSet) : Template
Recognize (probe : Features) : SimilarityVector
1
RecognitionSystem
Enroll (f : FeaturesSet)
TrainFeatureExtractor (trainingSet : FaceSet)
TrainRecognizer (trainingSet : FaceSet)
TestRecognition (testSet : FaceSet) : SimilarityMatrix

Figure 4.8: Class Diagram

4.2.6 Component Diagram

Component diagram is UML structure diagram which shows component and dependencies between the component. Model diagrams allow to show different views of a system, for example, as multi-layered (aka multi-tiered) application - multi-layered application model. Following component diagram shows the components of the proposed system, hierarchy of components and their relationships with each other.

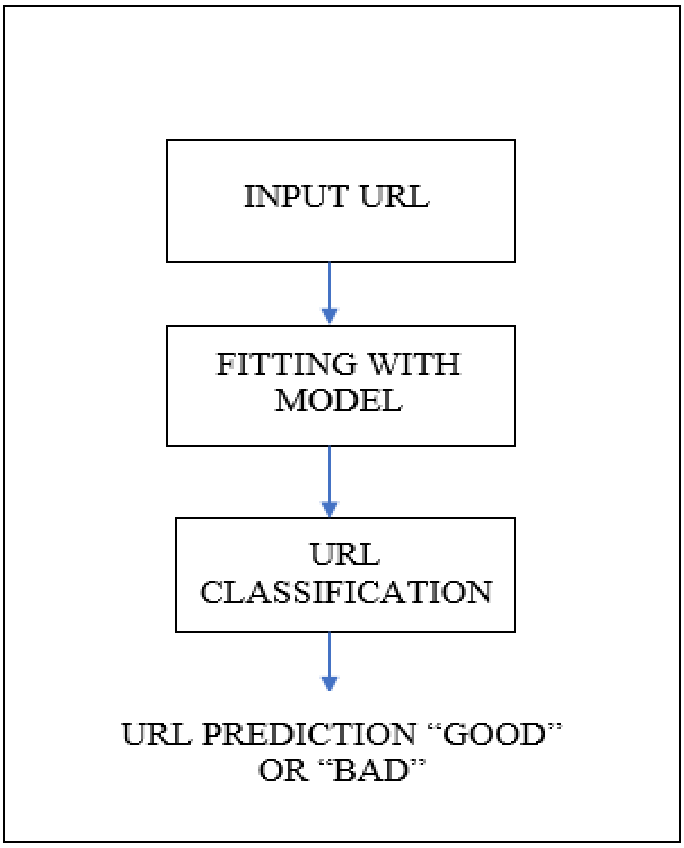
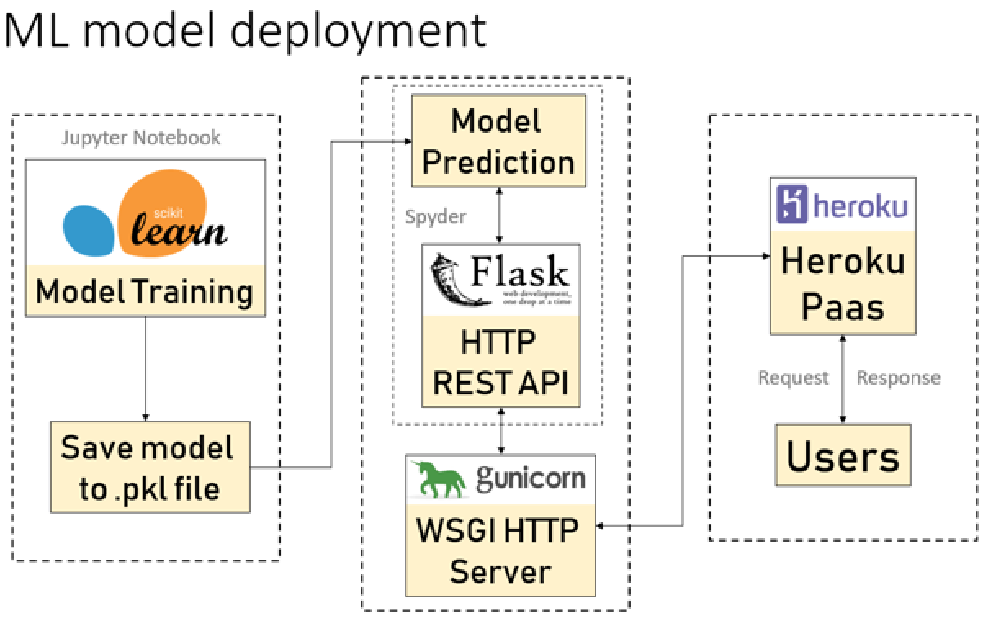


Figure 4.2.6 Component Diagram

4.2.7 Deployment Diagram

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.


ML model deployment

4.2.8 Sequence Activity Diagram Of Phishing

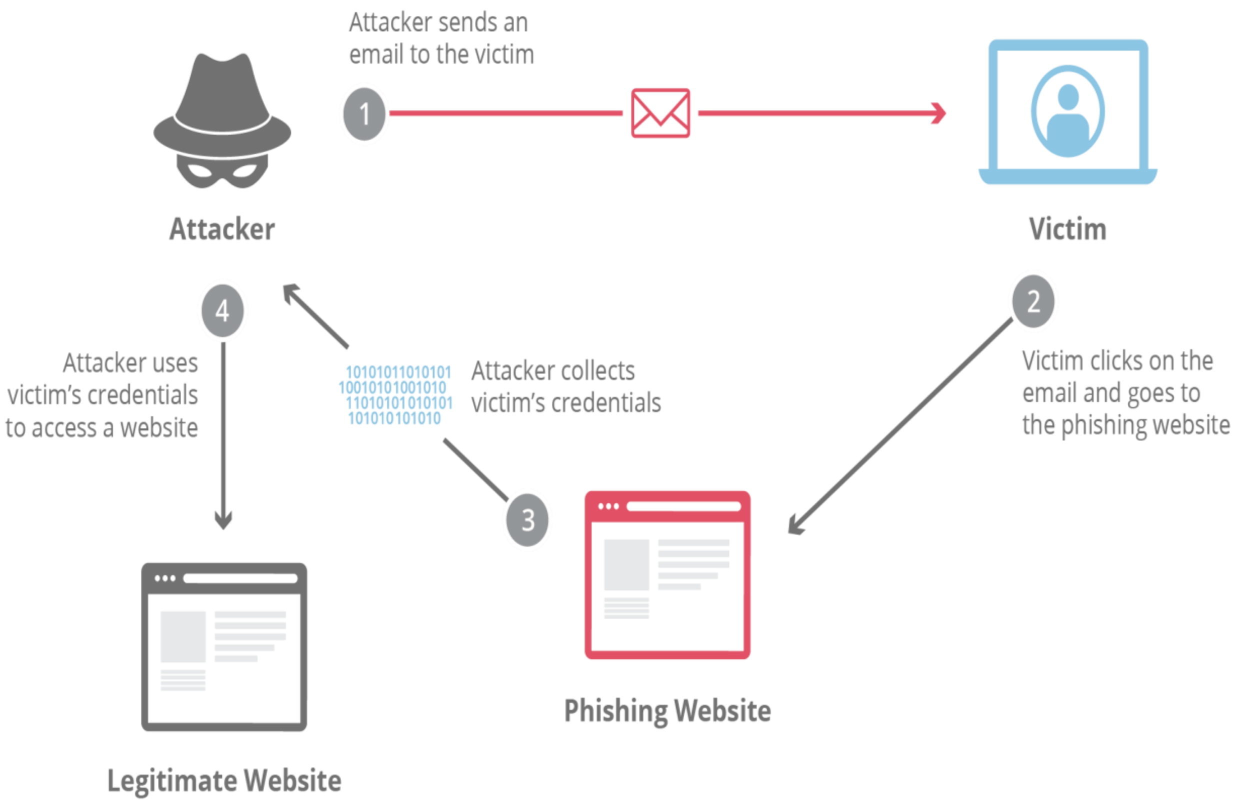
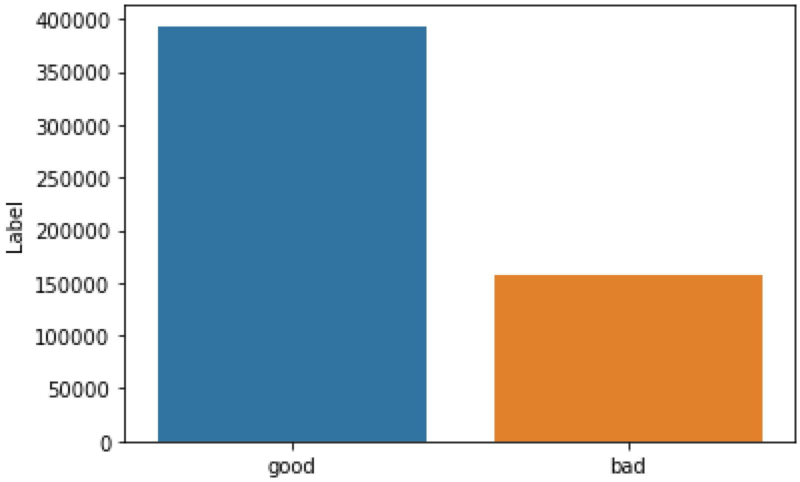

Attacker sends an email to the victim
1
Attacker
Victim
2
Victim clicks on the email and goes to the phishing website
4
Attacker uses victim's credentials to access a website
10101011010101
10010101001010
11010101010101
101010101010
Attacker collects victim's credentials
3
Phishing Website
Legitimate Website

Figure 4.2.8 Sequence activity diagram of Phishing

4.2.9 Bar graph (showing total no of Good and Bad websites in our dataset)



Here we can see there are aprox. 390000 good websites and approx. 160000 bad websites

4.3 Summery

Detailed design of project has been described in this chapter. Conclusion and future work of the project has been explained in the next chapter

Chapter 5

Conclusion and Result

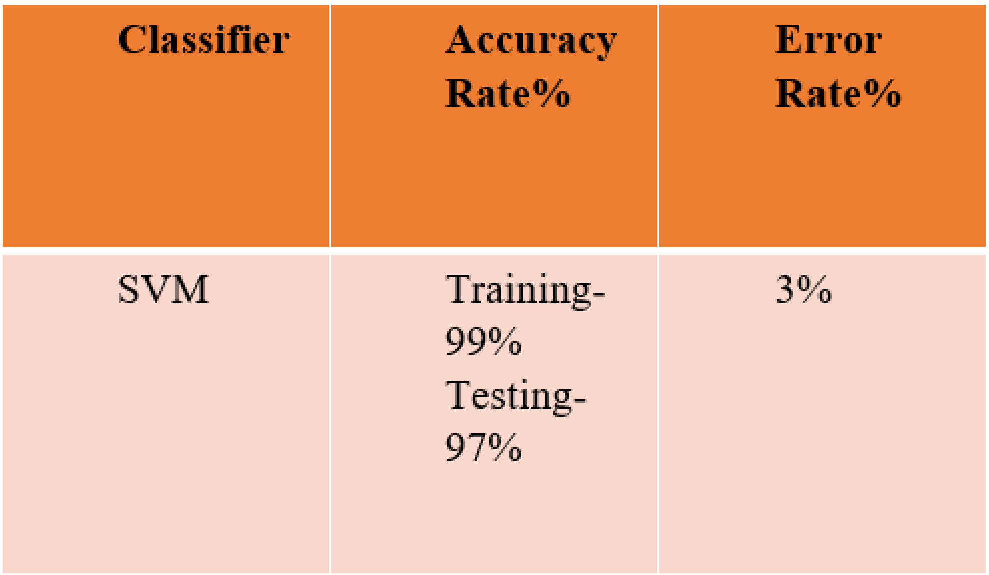
5.1 Conclusion

- To summarize, we have seen how phishing is a huge threat to the security and safety of the web and how phishing detection is an important problem domain. We have tested SVM machine learning algorithm on the 'Phishing Websites Dataset' and reviewed their results. We then built a Chrome extension for detecting phishing web pages. The extension allows easy deployment of our phishing detection model to end users. We have detected phishing websites using Support Vector Machine with and accuracy of 97.94%.

- For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction.

5.2 Result

- Scikit-learn tool has been used to import Machine learning algorithms. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers.
- Performance of classifiers has been evaluated by calculating classifier's accuracy score.
- Here we got model accuracy of 97% which is quite good.
  So Support Vector Machine is ideal to use.



Classifier	Accuracy Rate%	Error Rate%
SVM	Training-99% Testing-97%	3%

5.3 Bibliography

- Detection of phishing websites using machine learning International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV10IS050235 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 10 Issue 05, May-2021

- Joby James, Sandhya L, Ciza Thomas Detection of phishing websites using machine learning techniques. 2013 International Conference on Control Communication and Computing (ICCC)https://www.researchgate.net/publication/269032183
- Mustafa AYDIN, Nazife BAYKAL (2015) Feature extraction and classification phishing websites based on URL. IEEE
- WWW.geeksforgeeks.com