A

Major Project

On

**EFFICIENT EMAIL PHISHING DETETCTION USING MACHINE LEARNING**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

ANUGNA LINGAMPALLI (207R1A0590)

Under the Guidance of

**RAKSHITHA OKALI**

(Assistant Professor)

##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CMR TECHNICAL CAMPUS UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2020-2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



#### CERTIFICATE

This is to certify that the project entitled **“EFFICIENT EMAIL PHISHING DETECTION USING MACHINE LEARNING”** being submitted by **ANUGNA LINGAMPALLI (207R1A0590),** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Rakshitha Okali Dr. A. Raji Reddy**

(Assistant Professor) DIRECTOR

INTERNAL GUIDE

**Dr. K. Srujan Raju EXTERNAL EXAMINER**

HOD

**Submitted for viva voice Examination held on**

##### ACKNOWLEDGEMENT

Apart from my efforts, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I take this opportunity to express my profound gratitude and deep regard to my guide **Rakshitha Okali,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

I am obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. I also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. I am grateful for their constant support and help.

Finally, I would like to take this opportunity to thank my family for their constant encouragement, without which this assignment would not be completed. I sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

###### ANUGNA LINGAMPALLI (207R1A0590)

##### ABSTRACT

Emails are frequently utilized as a way of personal and professional communication. Banking information, credit reports, login data, and other sensitive personal information are commonly transmitted over email. This makes them valuable to cybercriminals, who can exploit the knowledge for their gain. Phishing is a technique used by con artists to steal sensitive information from people by impersonating well-known sources. The sender of a phished email can persuade you to disclose personal information under pretenses. The detection of a phished email is treated as a classification problem in this research, and this project shows how machine learning methods are used to categorize emails as phished or not. SVM classifier attains a maximum accuracy of 0.998 percent in email classification.

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 4.1 | Architecture for Efficient Email phishing Detection  using Machine learning | 11 |
| Figure 4.2 | Use Case Diagram for  Efficient Email phishing Detection using Machine learning | 12 |
| Figure 4.3 | Class Diagram for Efficient Email phishing Detection using Machine learning | 13 |
| Figure 4.4 | Sequence diagram for  Efficient Email phishing Detection using Machine learning | 14 |
| Figure 4.5  Figure 4.6  Figure 4.7 | Activity diagram for Efficient  Email phishing Detection using Machine learning  Collaboration Diagram for  Efficient Email phishing Detection using Machine learning  Deployment Diagram for  Efficient Email phishing Detection using Machine learning | 15  16  17 |

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO**. |
| Screenshot 6.1 | Training the Dataset | 25 |
| Screenshot 6.2 | Run Project | 25 |
| Screenshot 6.3 | Web Page View | 26 |
| Screenshot 6.4 | Admin Login | 26 |
| Screenshot 6.5  Screenshot 6.6  Screenshot 6.7  Screenshot 6.8  Screenshot 6.9  Screenshot 6.10  Screenshot 6.11  Screenshot 6.12  Screenshot 6.13 | Successful Admin Login  Uploading Dataset  Class Labels and Email Messages of Uploaded Dataset  Random Forest Output Screen for Uploaded Dataset  Spam Detection Screen  Input Message  Detection Result Screen  Other Input Message  Detection Result Screen | 27  27  28  28  29  29  30  30  31 |

###### ABSTRACT i

LIST OF FIGURES ii

LIST OF SCREENSHOTS iii

1. [INTRODUCTION](#_TOC_250030) 1

1.1 [PROJECT SCOPE 1](#_TOC_250029)

1.2 [PROJECT PURPOSE 1](#_TOC_250028)

1.3 [PROJECT FEATURES 1](#_TOC_250027)

1. LITERATURE SURVEY 2

2.1 TABLE OF SURVEY 2

2.2 PHISHING ENVIRONMENTS, TECHNIQUES,

AND COUNTER MEASURES: A SURVEY 4

2.3 EXPLORING SUSCEPTIBILITY TO PHISHING

IN THE WORKPLACE 4

2.4 INTELLIGENT DEEP MACHINE LEARNING

CYBER PHISHING URL DETECTION BASED 5

ON BERT FEATURES EXTRACTION

2.5 MACHINE LEARNING TECHNIQUES FOR

DETECTION OF WEBSITE PHISHING: A 6

REVIEW FOR PROMISES AND CHALLENGES

2.6 PHIBOOST – A NOVAL PHISHING

DETECTION MODEL USING ADAPTIVE 7

BOOSTING APPROACH

1. [SYSTEM ANALYSIS](#_TOC_250026) 8

3.1 [PROBLEM DEFINITION](#_TOC_250025) 8

3.2 [EXISTING SYSTEM](#_TOC_250024) 8

3.2.1 LIMITATIONS OF THE EXISTING SYSTEM 9

3.3 [PROPOSED SYSTEM](#_TOC_250023) 9

3.3.1 ADVANTAGES OF PROPOSED SYSTEM 10

3.4 [FEASIBILITY STUDY](#_TOC_250022) 10

3.4.1 [ECONOMIC FEASIBILITY](#_TOC_250021) 10

3.4.2 [TECHNICAL FEASIBILITY](#_TOC_250020) 11

3.4.3 SOCIAL FEASIBILITY 11

3.5 [HARDWARE & SOFTWARE REQUIREMENTS](#_TOC_250019) 11

3.5.1 [HARDWARE REQUIREMENTS](#_TOC_250018) 11

3.5.2 [SOFTWARE REQUIREMENTS](#_TOC_250017) 12

1. [ARCHITECTURE](#_TOC_250016)  13

4.1 [PROJECT ARCHITECTURE](#_TOC_250015) 13

4.2 [DESCRIPTION](#_TOC_250014) 14

4.3 [USE CASE DIAGRAM](#_TOC_250013) 14

4.4 [CLASS DIAGRAM](#_TOC_250012)  15

4.5 [SEQUENCE DIAGRAM](#_TOC_250011) 16

4.6 [ACTIVITY DIAGRAM 1](#_TOC_250010)7

4.7 COLLABORATION DIAGRAM 18

4.8 DEPLOYMENT DIAGRAM 19

1. IMPLEMENTATION 20

5.1 SAMPLE CODE 20

1. SCREENSHOTS 27
2. [TESTING 34](#_TOC_250009) 7.1 [INTRODUCTION TO TESTING 3](#_TOC_250008)4

7.2 [TYPES OF TESTING 3](#_TOC_250007)4

7.2.1 UNIT TESTING 34

7.2.2 [INTEGRATION TESTING 3](#_TOC_250006)5

7.2.3 [FUNCTIONAL TESTING 3](#_TOC_250005)5

7.2.4 SYSTEM TESTING 36

7.2.5 WHITE BOX TESTING 36

7.2.6 BLACK BOX TESTING 36

7.2.7 ACCEPTANCE TESTING 37

7.3 [TEST CASES 38](#_TOC_250004)

7.3.1 [CLASSIFICATION 3](#_TOC_250003)8

1. CONCLUSION & FUTURE SCOPE 39

8.1 PROJECT CONCLUSION 39

8.2 [FUTURE SCOPE](#_TOC_250002) 40

##### BIBLIOGRAPHY 41

9.1 [REFERENCES](#_TOC_250001) 41

9.2 [GITHUB LINK](#_TOC_250000) 42

# 1. INTRODUCTION

#### 1. INTRODUCTION

##### 1.1 PROJECT SCOPE

The scope of efficient email phishing detection using machine learning is significant and continually evolving as cyber threats become more sophisticated. Phishing attacks remain a prevalent and effective method for cybercriminals to gain unauthorized access to sensitive information, compromise systems, and carry out malicious activities. Efficient email phishing detection using machine learning plays a crucial role in mitigating these risks.

##### 1.2 PROJECT PURPOSE

The purpose of efficient email phishing detection using machine learning is to enhance Cyber Security by identifying and mitigating phishing threats in email communications. Phishing is a prevalent form of cyber attack where attackers attempt to deceive individuals into divulging sensitive information, such as login credentials or financial details, by posing as a trustworthy entity.

##### 1.3 PROJECT FEATURES

Efficient email phishing detection using machine learning relies on various features to accurately identify and classify phishing threats. These features are extracted from different aspects of emails and user behavior.

some of the essential features are :

* Sender Information
* IP Address Analysis
* Email Header Analysis
* Content Analysis
* User Analysis
* Cross-Validation

## 2. LITERATURE SURVEY

##### 2. LITERATURE SURVEY

**2.1 Table Of Survey**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Author Name** | **Year** | **Methodology Used** | **Algorithms Used** | **Description** |
| 1 | Chandra  sekaran et al. | 2006 | Dependence on distinctive structural features of emails, in cooperation with SVM | SVM | Detecting phishing emails by utilizing email structural features combined with SVM |
| 2 | Gansterer & Polz | 2009 | Development of a filtering system using various classifiers to categorize emails | Various classifiers | Creation of a filtering system for categorizing emails into legitimate, spam, and phishing categories |
| 3 | Wu et al. | 2010 | Development of a sender authentication protocol (SAP) for spoofing emails in Microsoft OutlookTM | Sender Authentication Protocol (SAP) | Introduction of a sender authentication protocol for Microsoft OutlookTM to prevent spoofing emails |
| 4 | Azad | 2011 | Testing of existing algorithms (Naive Bayes, logistic regression, SVM) with feature selection | Naive Bayes, Logistic Regression, Support Vector Machine (SVM) | Testing various algorithms for phishing email detection with feature selection. |

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | Pandey & Ravi | 2012 | Application of various classification approaches for phishing email detection with feature selection | Multilayer Perceptron (MLP), Decision Trees (DT), Support Vector Machine (SVM), Genetic Programming (GP), Logistic Regression (LR) | Testing of different classification approaches for phishing email detection with feature selection |
| 6 | Nizamani et al. | 2014 | Comparison of fraudulent email detection rate among different categories | SVM, NB, J48, CCM | Comparison of fraudulent email detection rates among different categories using various classifiers |
| 7 | Kathirvalavakumar et al. | 2015 | Proposal of a multilayer neural network with feedforward pruning algorithm for phishing email detection | Multilayer Neural Network | Introduction of a multilayer neural network with feedforward pruning algorithm for phishing email detection |

3

**2.2 PHISHING ENVIRONMENTS, TECHNIQUES, AND COUNTER MEASURES: A SURVEY**

Phishing has become an increasing threat in online space, largely driven by the evolving web, mobile, and social networking technologies. Previous phishing taxonomies have mainly focused on the underlying mechanisms of phishing but ignored the emerging attacking techniques, targeted environments, and countermeasures for mitigating new phishing types. This survey investigates phishing attacks and anti-phishing techniques developed not only in traditional environments such as e-mails and websites, but also in new environments such as mobile and social networking sites. Taking an integrated view of phishing, we propose a taxonomy that involves attacking techniques, countermeasures, targeted environments and communication media.

The taxonomy will not only provide guidance for the design of effective techniques for phishing detection and prevention in various types of environments, but also facilitate practitioners in evaluating and selecting tools, methods, and features for handling specific types of phishing problems. This study reveals that anti-phishing research and development has focused on phishing in e-mails and websites, but paid little attention to that in IM, social networks, voice, blogs and web forums; further, phishing in mobile communication has yet to be explored from the technical perspective.

**2.3 EXPLORING SUSCEPTIBILITY TO PHISHING IN THE WORKPLACE**

Phishing emails provide a means to infiltrate the technical systems of organizations by encouraging employees to click on malicious links or attachments. Despite the use of awareness campaigns and phishing simulations, employees remain vulnerable to phishing emails. The present research uses a mixed methods approach to explore employee susceptibility to targeted phishing emails, known as spear phishing. In study one, nine spear phishing simulation emails sent to 62,000 employees over a six-week period were rated according to the presence of authority and urgency influence techniques. Results demonstrated that the presence of authority cues increased the likelihood that a user would click a suspicious link contained in an email.

4

In study two, six focus groups were conducted in a second organization to explore whether additional factors within the work environment impact employee susceptibility to spear phishing. We discuss these factors in relation to current theoretical approaches and provide implications for user communities. Work based norms and routines likely represent a primary factor impacting response behaviour within the workplace, influencing the development of context-specific habits, expectations and perceptions of risk. Reflective of the combined findings of Study One and Two, considering aspects of the email that is received, the individual who receives it, and the context in which it is encountered, within theoretical approaches is vital if susceptibility within the workplace is to be truly understood. It is hoped that the findings of the current study will provide a basis for further theoretical development in this field, whilst also presenting an initial aid for user communities to consider, and begin to address, the range of potential susceptibility factors that may be present within organizational settings.

**2.4 INTELLIGENT DEEP MACHINE LEARNING CYBER PHISHING URL DETECTION BASED ON BERT FEATURES EXTRACTION**

Recently, phishing attacks have been a crucial threat to cyberspace security. Phishing is a form of fraud that attracts people and businesses to access malicious uniform resource locators (URLs) and submit their sensitive information such as passwords, credit card ids, and personal information. Enormous intelligent attacks are launched dynamically with the aim of tricking users into thinking they are accessing a reliable website or online application to acquire account information. Researchers in cyberspace are motivated to create intelligent models and offer secure services on the web as phishing grows more intelligent and malicious every day.

In this project, a novel URL phishing detection technique based on BERT feature extraction and a deep learning method is introduced. BERT was used to extract the URLs’ text from the Phishing Site Predict dataset. Then, the natural language processing (NLP) algorithm was applied to the unique data column and extracted a huge number of useful data features in terms of meaningful text information. Next, a deep convolutional neural network method was utilised to detect phishing URLs. It was used to constitute words or n-grams in order to extract higher-level features. Then, the data were classified into legitimate and phishing URLs.

5

The experiments showed that the proposed method had achieved 96.66% accuracy in the results, and then the obtained results were compared to other literature review works. The results showed that the proposed method was efficient and valid in detecting phishing websites’ URLs.

**2.5 MACHINE LEARNING TECHNIQUES FOR DETECTION OF WEBSITE PHISHING: A REVIEW FOR PROMISES AND CHALLENGES**

Websites phishing is a cyber-attack that targets online users to steal their sensitive information including login credentials and banking details. Attackers fool the users by presenting the masked webpage as a legitimate or trustworthy to retrieve their essential data. Several solutions to phishing websites attacks have been proposed such as heuristics, blacklist or whitelist, and Machine Learning (ML) based techniques. This paper presents the state of art techniques for detection of phishing websites using the ML techniques. This research identifies solutions to the website's phishing problem based on the ML techniques.

The majority of the examined approaches are focused on traditional ML techniques. Random Forests (RFs), Support Vector Machines (SVMs), Naïve Bayes (NB), and Ada Boosting are the powerful ML models examined in the literature. This survey paper also identifies deep learning-based techniques to demonstrate better performance for detecting phishing websites compared to the conventional ML techniques. Challenges to ML techniques identified in this work include over fitting, low accuracy, and ML techniques' ineffectiveness in case of unavailability of enough training data.

This research suggests that Internet users should know about phishing to avoid cyber-attacks. Phishing attacks present negative impacts on web owners and end-users. The reputation of website owners becomes questionable when attackers launch an attack, and as a result of the attack, website users lose their sensitive information

6

**2.6 PHIBOOST – A NOVAL PHISHING DETECTION MODEL USING ADAPTIVE BOOSTING APPROACH**

Every day, cyberattacks increase and use different strategies. One of the most common cyberattacks is Phishing, where the attacker collects sensitive and confidential information by pretending as a trusted party.

Different traditional strategies have been introduced for anti-phishing, such as blacklisted, heuristic search and visual similarity. Most of these traditional methods have a high false rate and take a long time to detect the phishing website. New modes have been introduced using machine learning techniques which improve the detection’s accuracy. Machine learning techniques require a huge amount of data called features that are collected from different websites. These collected features are classified into four categories.

This project introduces a novel detection model by utilizing features’ selection to pick up the highly correlated features with the class label. The phase of features’ selection employs independent significance features library from MATLAB and heat-map from Python to find the highly correlated features. The results of this study explored the best splitting rate for the dataset to train the machine learning model, which was 70%. Then, the proposed model uses an adaptive boosting approach which consists of multiple classifiers to increase the model’s accuracy. The proposed model produces an extremely high predictive accuracy of approximately 99%.

7

2

## 3. SYSTEM ANALYSIS

##### 3 . SYSTEM ANALYSIS

**SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

##### 3.1 PROBLEM DEFINITION

##### Efficient email phishing detection using machine learning refers to the application of advanced computational techniques, particularly machine learning algorithms, to automatically identify and classify phishing attempts within email communications. The goal is to create a robust and adaptive system that can analyze various features of emails to distinguish between legitimate messages and those that aim to deceive recipients for malicious purposes.

##### 3.2 EXISTING SYSTEM

The attackers add sub domains to the links to make them appear authentic. The number of dots in the link rose as sub domains were added. As suggested by In a valid email, the number dots should not be used. More than three [three]. This is a binary feature, meaning it determines whether or not a link exists. It would be in the mail if the number of dots was more prominent than three. This is a phished email. The total number of links is: In general, phishing emails provide more information. In comparison to ham, the transmitter attempts to send many links. By tricking the user, you might direct him to an illicit website. This is a recurring feature.

8

###### 3.2.1 DISADVANTAGES OF EXISTING SYSTEM

* An existing system not implemented an effective ML Classifiers like SVM,RF,NB.
* An existing system not implemented for large number of datasets.

##### 3.3 PROPOSED SYSTEM

The designing of innovative models and architectures are transforming this combination of information along with user requirements for giving improved well-versed decisions and enriching the customer experience. The vast volume of information within the communicating networks fetches vital challenges for assuring security basics within a disseminated setting. Block Chain, originally launched as a Bit Coin Crypto Currency by Satoshi Nakamoto, has advanced beyond that. It facilitates a reliable platform for exchanging services and transactions through a disseminated network.

The presence of JavaScript in an email indicates that the sender is either trying to conceal information or activate specific browser changes. This is a one-of-a-kind feature. The presence of the script> tag in an email indicates that it has been phished. Form tag: Phishing emails feature forms integrated into them to acquire information from users. This is a binary characteristic, meaning that the presence of a form tag indicates that the email is phished. HTML emails allow the sender to include embedded graphics and URLs, which are not possible with plain text emails. If the email has an HTML tag, it is considered phishing. This is a one-of-a-kind feature. The use of action words in emails shows if the sender expects the recipient to do a specific action, such as clicking on a link, filling out a form, or submitting detailed information. This is a recurring feature.

The word PayPal indicates that the sender is posing as a member of a recognized organization. The word "PayPal" appears in the mail's links or the "from" section, implying that the sender is affiliated with PayPal. This is a one-of-a-kind feature.The presence of the term bank is a binary indicator indicating the message is about banking. Either the sender is posing as a member of the financial organization, or the reader is looking at the reader's credentials. The word account appears in the email, indicating that it seeks emails tied to an account. It could be a social media account, a bank account, or something else entirely. It's a one of- a-kind feature.

9

###### 3.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

* SVM is a supervised technique often used for text categorization because of its speed and accuracy. It generates a hyper\_plane, a two-dimensional line that best separates the categories, based on the training data. The decision boundary is the name given to this hyper\_plane
* The Naive Bayes classifier[20] is a probabilistic technique that uses the Bayes theorem to classify sample data.

##### 3.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.Three key considerations involved in the feasibility analysis are,

* Economic Feasibility
* Technical Feasibility
* Social Feasibility

###### 3.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

10

###### 3.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**3.4.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

##### 3.5 HARDWARE & SOFTWARE REQUIREMENTS

###### 3.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

* Processor - Pentium –IV
* RAM - 4 GB (min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

11

##### 3.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

* Operating system **:** Windows 7 Ultimate.
* Coding Language **:** Python.
* Front-End  **:** Python.
* Back-End **:** Django-ORM
* Designing **:** Html, CSS, Javascript.
* Data Base **:** MySQL (WAMP Server).

12

## 4. ARCHITECTURE

##### 4. ARCHITECTURE

##### 4.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

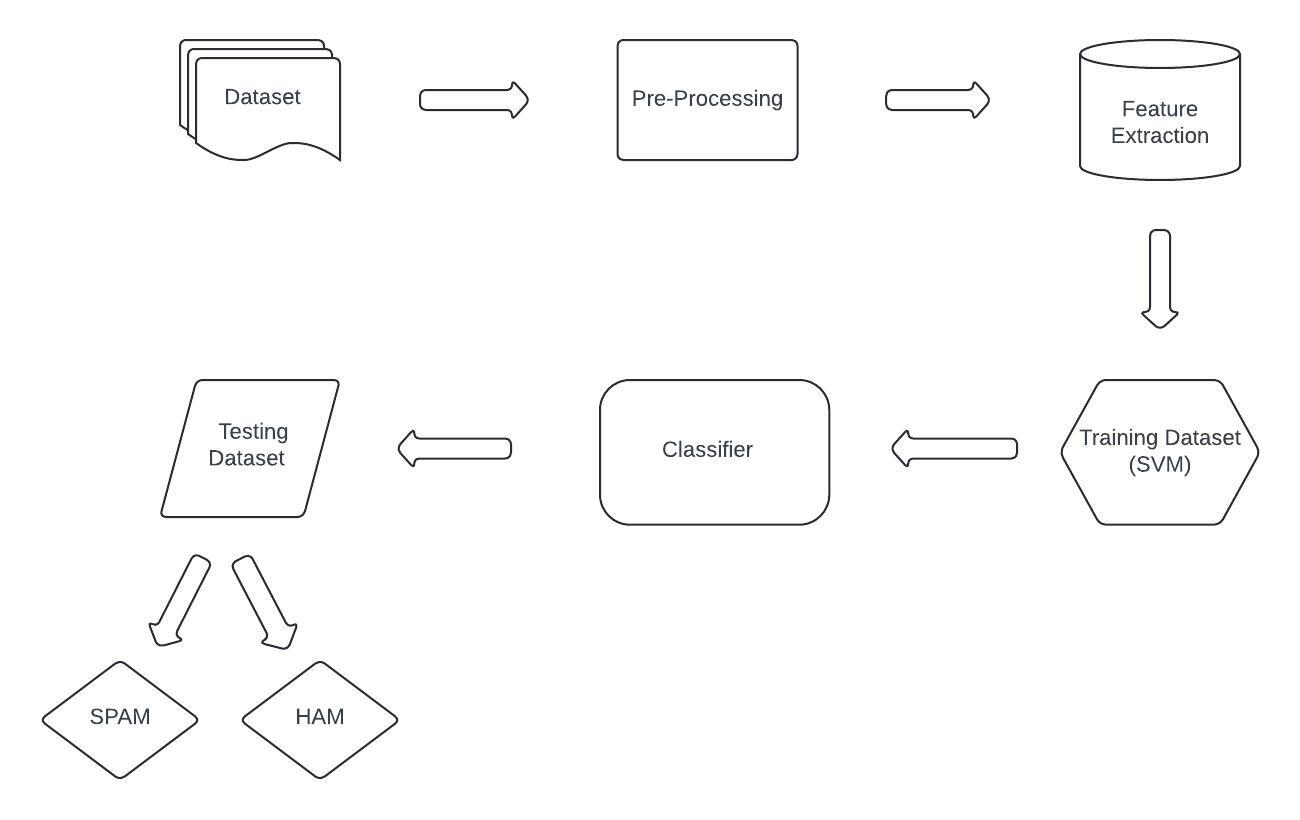


Figure 4.1: Project Architecture of Efficient Email Phishing Detection using Machine Learning

13

###### 4.2 DESCRIPTION

With the escalating sophistication of phishing attacks, traditional email security measures are becoming increasingly insufficient. This paper introduces a novel solution for bolstering email security through the integration of machine learning techniques. By leveraging advanced algorithms and feature engineering, our approach aims to efficiently identify and thwart phishing attempts, providing a robust defense against evolving cyber threats

###### 4.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Figure 4.2: Use Case Diagram for Efficient Email Phishing Detection using Machine Learning

14

##### 

##### 4.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

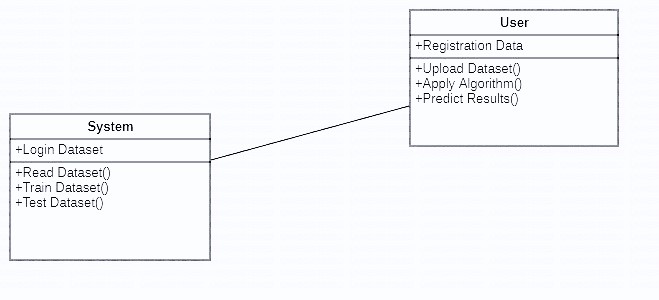


Figure 4.3: Class Diagram for Efficient Email Phishing Detection using Machine Learning

15

**4.5 SEQUENCE DIAGRAM**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

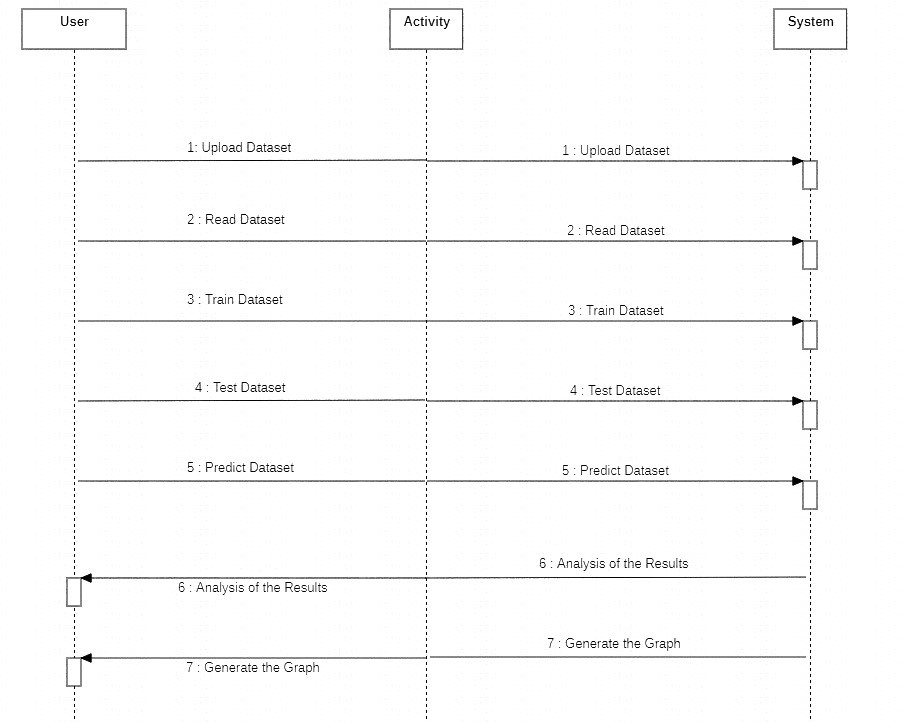


Figure 4.4: Sequence Diagram for Efficient Email Phishing Detection using Machine Learning

16

###### 4.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Figure 4.5: Activity Diagram for Efficient Email Phishing Detection using Machine Learning

17

**4.7 COLLABORATION DIAGRAM**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

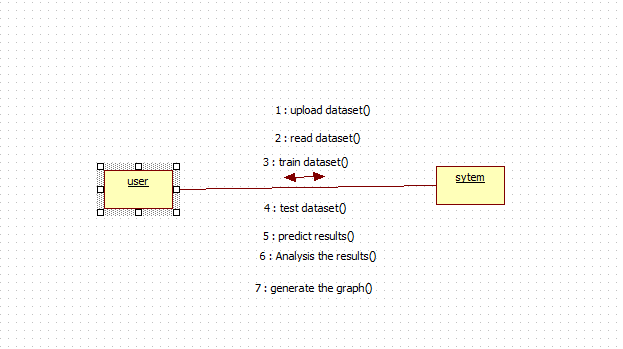


Figure 4.6: Collaboration Diagram of Efficient Email Phishing Detection using Machine Learning

18

**4.8 DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



Figure 4.7: Deployment Diagram of Efficient Email Phishing Detection using Machine Learning

19

## 5. IMPLEMENTATION

##### 5.1 SAMPLE CODE

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

import pymysql

from django.http import HttpResponse

from django.core.files.storage import FileSystemStorage

import os

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from string import punctuation

from nltk.corpus import stopwords

import nltk

from nltk.stem import WordNetLemmatizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics import accuracy\_score

import pickle

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from genetic\_selection import GeneticSelectionCV

from sklearn import linear\_model

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

20

global rf, tfidf\_vectorizer

def cleanPost(doc):

tokens = doc.split()

table = str.maketrans('', '', punctuation)

tokens = [w.translate(table) for w in tokens]

tokens = [word for word in tokens if word.isalpha()]

tokens = [w for w in tokens if not w in stop\_words]

tokens = [word for word in tokens if len(word) > 1]

tokens = [lemmatizer.lemmatize(token) for token in tokens]

tokens = ' '.join(tokens)

return tokens

def UploadDataset(request):

if request.method == 'GET':

return render(request, 'UploadDataset.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def SpamDetection(request):

if request.method == 'GET':

return render(request, 'SpamDetection.html', {})

def Login(request):

if request.method == 'GET':

return render(request, 'Login.html', {})

21

def UserLogin(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

if username == 'admin' and password == 'admin':

context= {'data':"Welcome "+username}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':'Invalid login details'}

return render(request, 'Login.html', context)

def UploadDatasetAction(request):

if request.method == 'POST':

file = request.FILES['t1']

dataset = pd.read\_csv("Dataset/spam\_ham\_dataset.csv",encoding='iso-8859-1',nrows=50)

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Class Label</font></th>'

output+='<th><font size=3 color=black>Email Message</font></th>'

for i in range(len(dataset)):

msg = dataset.get\_value(i, 'text')

label = dataset.get\_value(i, 'label')

output+='<tr><td><font size=3 color=black>'+str(label)+'</font></td>'

output+='<td><font size=3 color=black>'+msg+'</font></td>'

output+="</table><br/><br/><br/><br/><br/><br/>"

context= {'data':output}

return render(request, 'ViewDataset.html', context)

22

def TrainDataGA(request):

if request.method == 'GET':

global rf, tfidf\_vectorizer

Y = np.load("model/Y.txt.npy")

X = np.load("model/X.txt.npy")

estimator = linear\_model.LogisticRegression(solver="liblinear", multi\_class="ovr") #BUILDING GENETIC ALGORITHM WITH NAME CALLED SELECTOR

selector = GeneticSelectionCV(estimator,

cv=5,

verbose=1,

scoring="accuracy",

max\_features=5,

n\_population=50,

crossover\_proba=0.5,

mutation\_proba=0.2,

n\_generations=10,

crossover\_independent\_proba=0.5,

mutation\_independent\_proba=0.05,

tournament\_size=3,

n\_gen\_no\_change=10,

caching=True,

n\_jobs=-1)

selector = selector.fit(X, Y)#OPTIMIZING FEATURES WITH GENETIC ALGORITHM OBJECT SELECTOR

print(selector.support\_)

X\_selected\_features = X[:,selector.support\_==True]#SELECTING IMPORTANT FEATURES

data = X[:,selector.support\_==True]#assiging all seleccted features to data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data, Y, test\_size=0.2, random\_state = 0)

23

rf = RandomForestClassifier() #now training random forest with selected features

rf.fit(data, Y)

predict = rf.predict(X\_test)

acc = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test,predict,average='macro') \* 100

r = recall\_score(y\_test,predict,average='macro') \* 100

f = f1\_score(y\_test,predict,average='macro') \* 100

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Algorithm Name</font></th>'

output+='<th><font size=3 color=black>Accuracy</font></th>'

output+='<th><font size=3 color=black>Precision</font></th>'

output+='<th><font size=3 color=black>Recall</font></th>'

output+='<th><font size=3 color=black>FScore</font></th></tr>'

output+='<tr><td><font size=3 color=black>Random Forest with Genetic Algorithm</font></td>'

output+='<td><font size=3 color=black>'+str(acc)+'</font></td>'

output+='<td><font size=3 color=black>'+str(p)+'</font></td>'

output+='<td><font size=3 color=black>'+str(r)+'</font></td>'

output+='<td><font size=3 color=black>'+str(f)+'</font></td>'

output+="</table><br/><br/><br/><br/><br/><br/>"

context= {'data':output}

return render(request, 'TrainData.html', context)

def TrainData(request):

if request.method == 'GET':

global rf, tfidf\_vectorizer

Y = np.load("model/Y.txt.npy")

X = np.load("model/X.txt.npy")

24

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state = 0)

with open('model/tfidf.txt', 'rb') as file:

tfidf\_vectorizer = pickle.load(file)

file.close()

if os.path.exists('model/rf.txt'):

with open('model/rf.txt', 'rb') as file:

rf = pickle.load(file)

file.close()

else:

rf = RandomForestClassifier()

rf.fit(X\_train, y\_train)

with open('model/rf.txt', 'wb') as file:

pickle.dump(rf, file)

file.close()

predict = rf.predict(X\_test)

acc = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test,predict,average='macro') \* 100

r = recall\_score(y\_test,predict,average='macro') \* 100

f = f1\_score(y\_test,predict,average='macro') \* 100

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Algorithm Name</font></th>'

output+='<th><font size=3 color=black>Accuracy</font></th>'

output+='<th><font size=3 color=black>Precision</font></th>'

output+='<th><font size=3 color=black>Recall</font></th>'

output+='<th><font size=3 color=black>FScore</font></th></tr>'

output+='<tr><td><font size=3 color=black>Random Forest</font></td>'

output+='<td><font size=3 color=black>'+str(acc)+'</font></td>'

output+='<td><font size=3 color=black>'+str(p)+'</font></td>'

output+='<td><font size=3 color=black>'+str(r)+'</font></td>'

output+='<td><font size=3 color=black>'+str(f)+'</font></td>'

output+="</table><br/><br/><br/><br/><br/><br/>"

25

context= {'data':output}

return render(request, 'TrainData.html', context)

def SpamDetectionAction(request):

if request.method == 'POST':

global rf, tfidf\_vectorizer

message = request.POST.get('t1', False)

msg1 = message.strip().lower()

clean = cleanPost(msg1)

tfidf = tfidf\_vectorizer.transform([clean]).toarray()

predict = rf.predict(tfidf)

predict = predict[0]

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Email Message</font></th>'

output+='<th><font size=3 color=black>Detection Result</font></th></tr>'

if predict == 0:

output+='<tr><td><font size=3 color=black>'+message+'</font></td><td><font size=3 color=black>HAM</td></tr>'

if predict == 1:

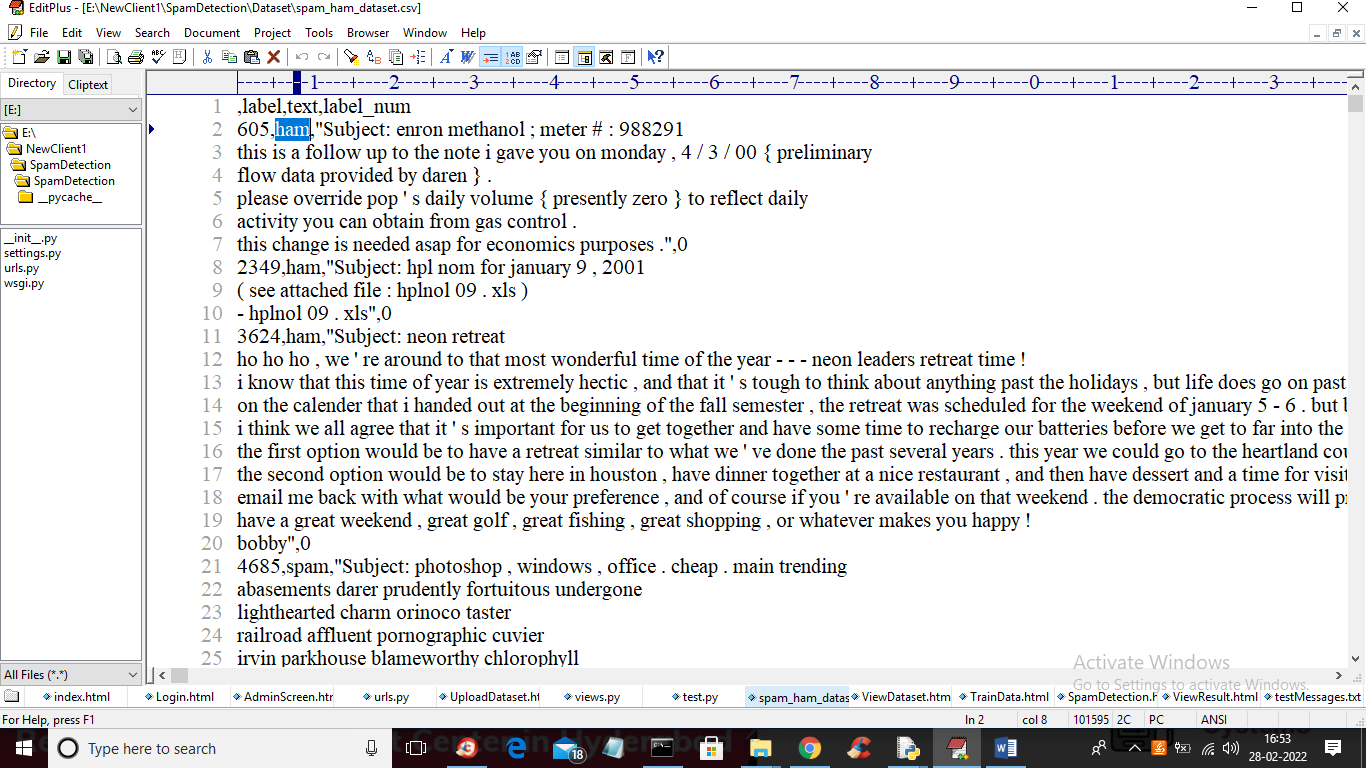
output+='<tr><td><font size=3 color=black>'+message+'</font></td><td><font size=3 color=black>SPAM</td></tr>'

context= {'data':output}

return render(request, 'ViewResult.html', context)

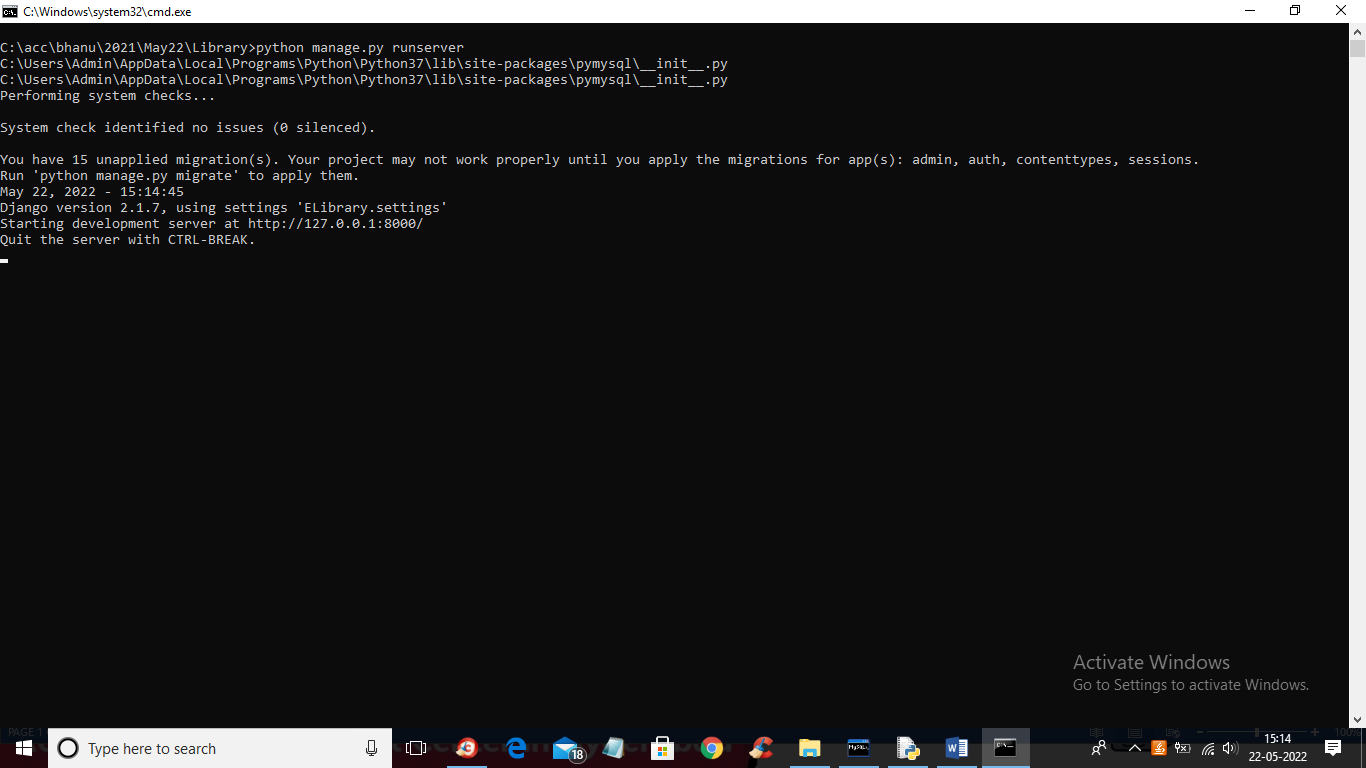
26

## 6. SCREENSHOTS

****

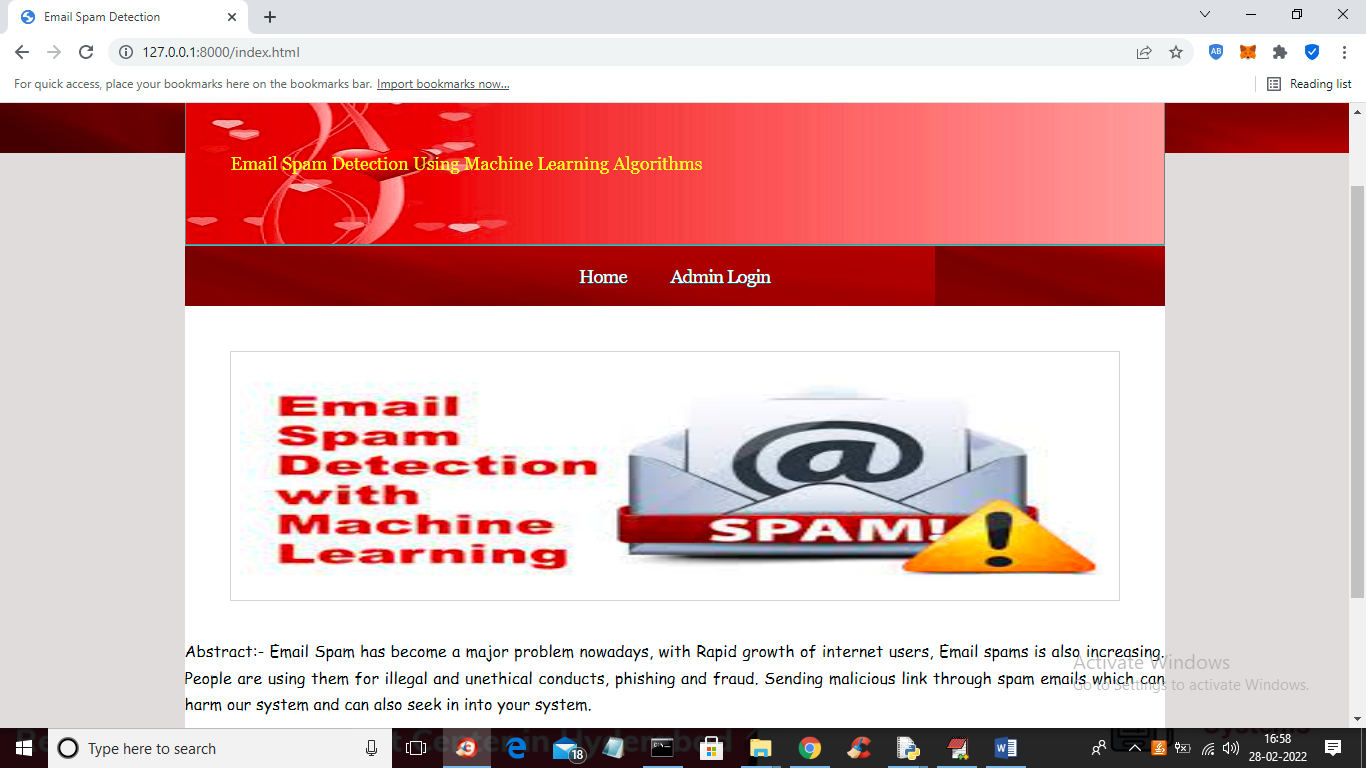
Screenshot 6.1 : Training the Dataset

In above screen first row represents dataset column names and remaining rows contains EMAIL message and class label as HAM or SPAM and by using above dataset we will train Random Forest algorithm.



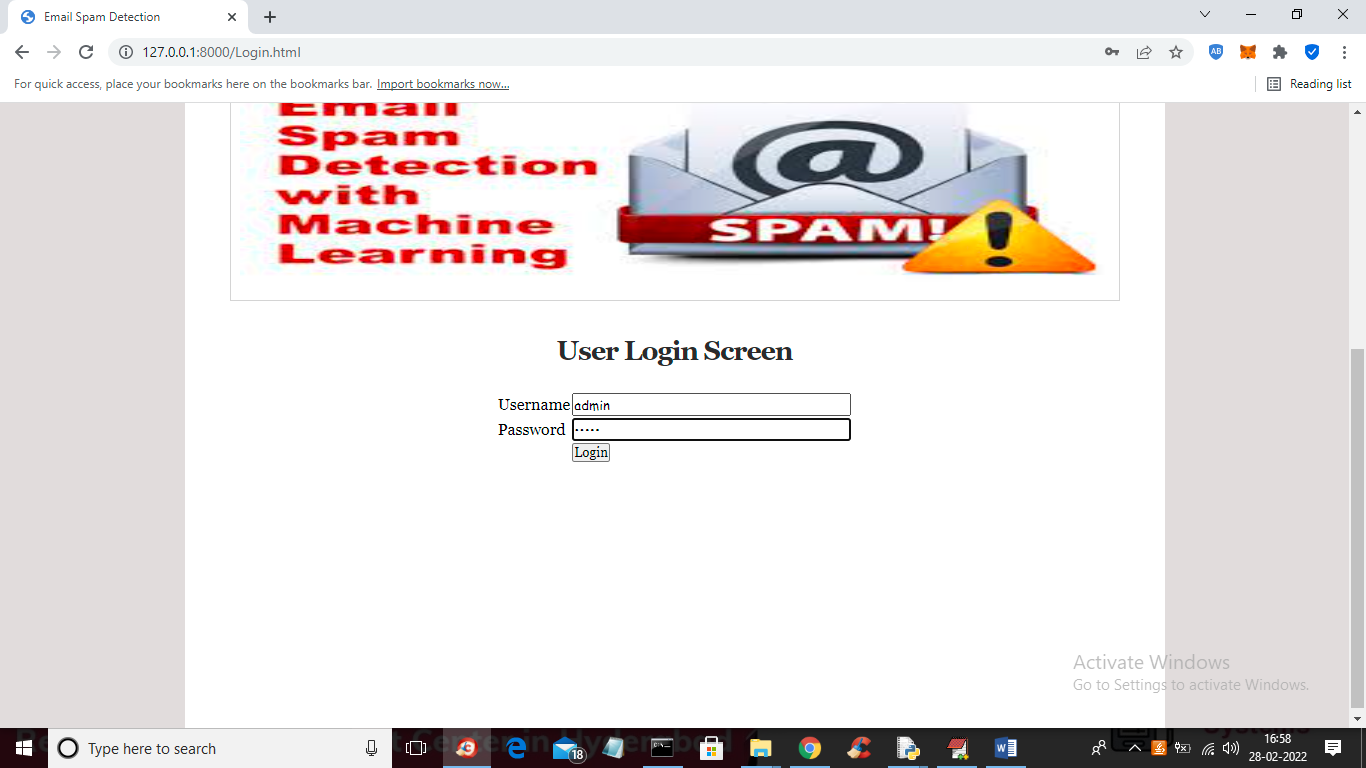
Screenshot 6.2 : Run the Project

Then open browser and enter URL as ‘http://127.0.0.1:8000/index.html’ and press enter key to get below screen



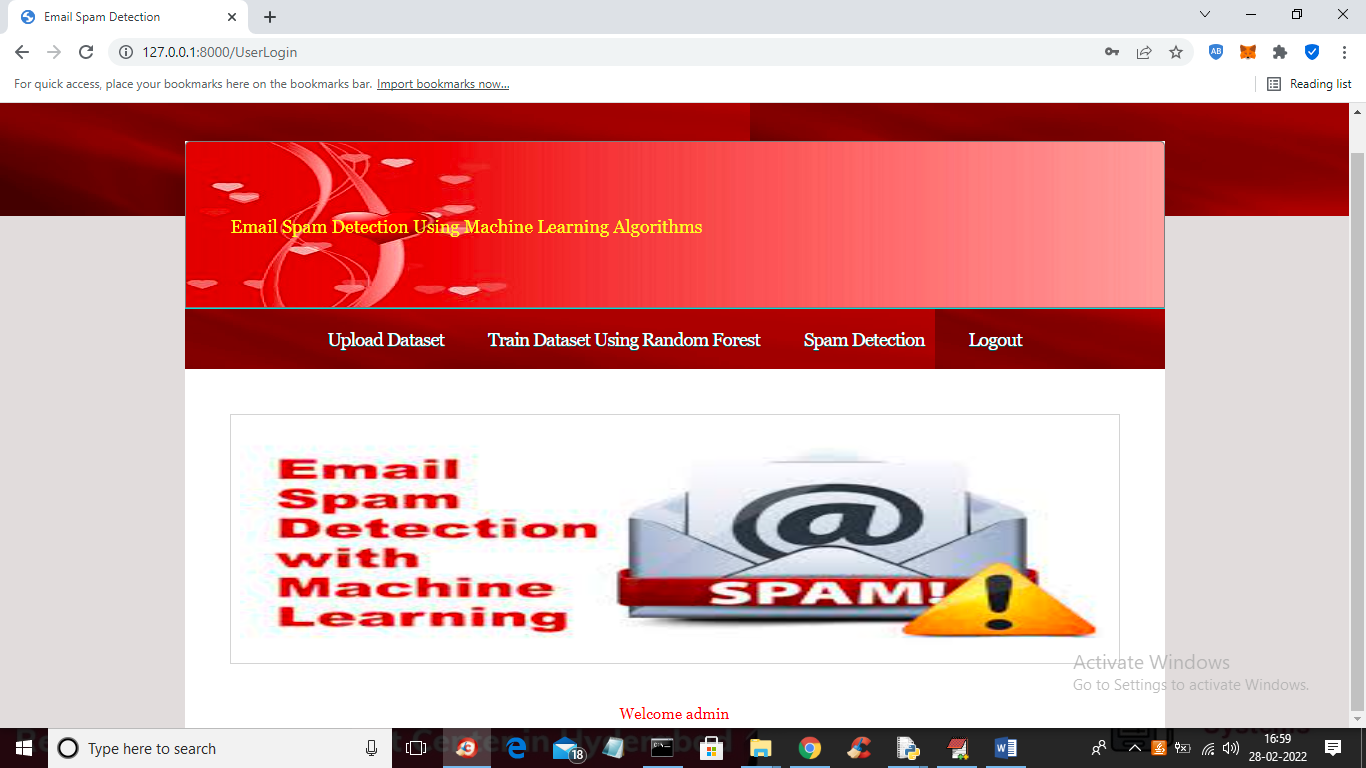
Screenshot 6.3 : Webpage View

Click on ‘Admin Login’ link to get below login screen



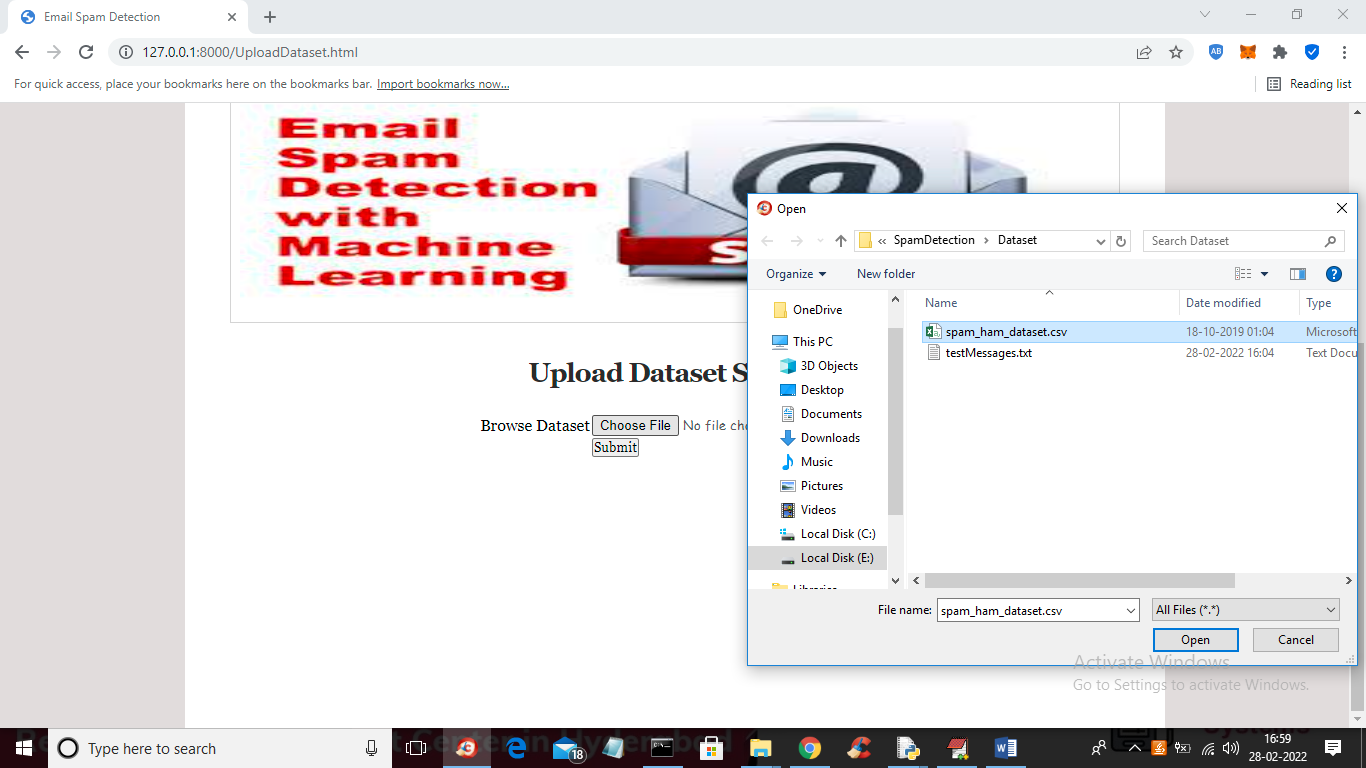
Screenshot 6.4 : Admin Login

Enter the details of the Username and Password and click on “Login”



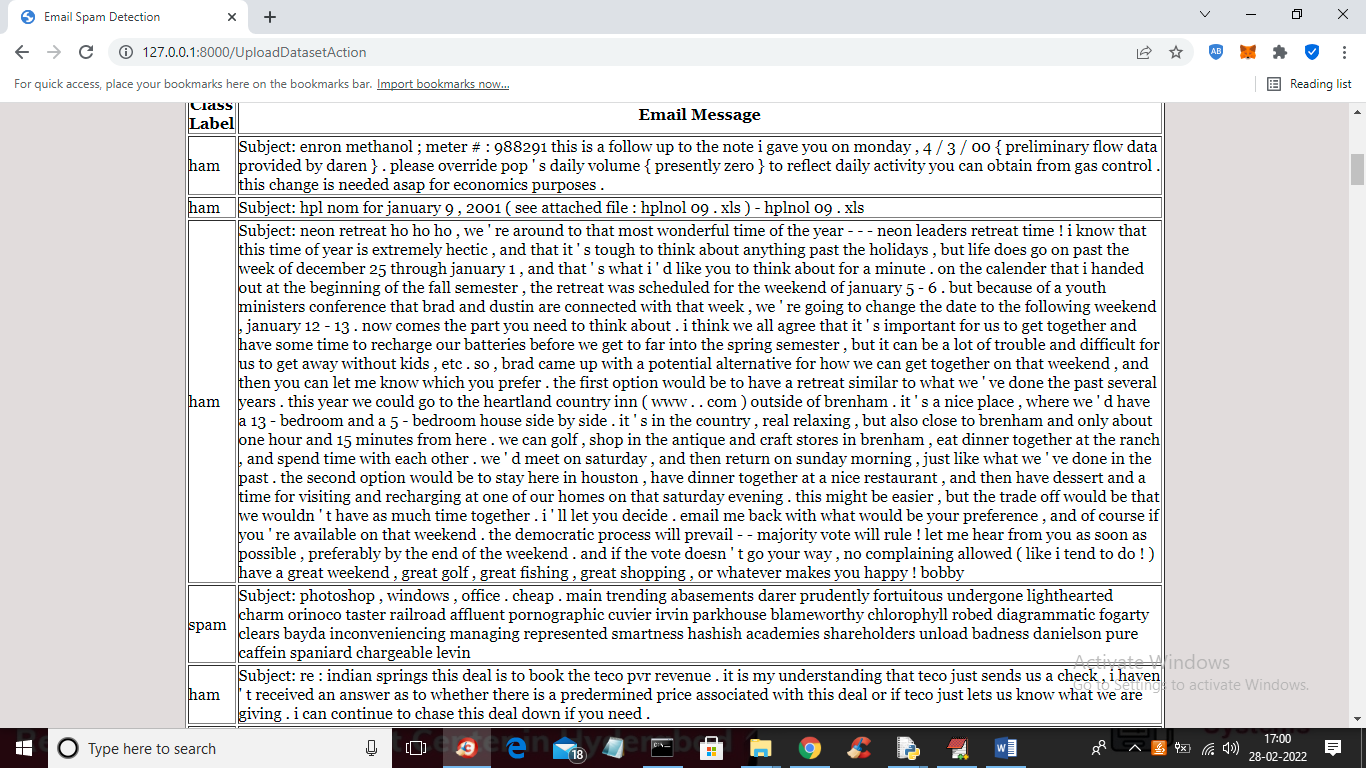
Screenshot 6.5 : Successful Admin Login

Click on Upload Dataset



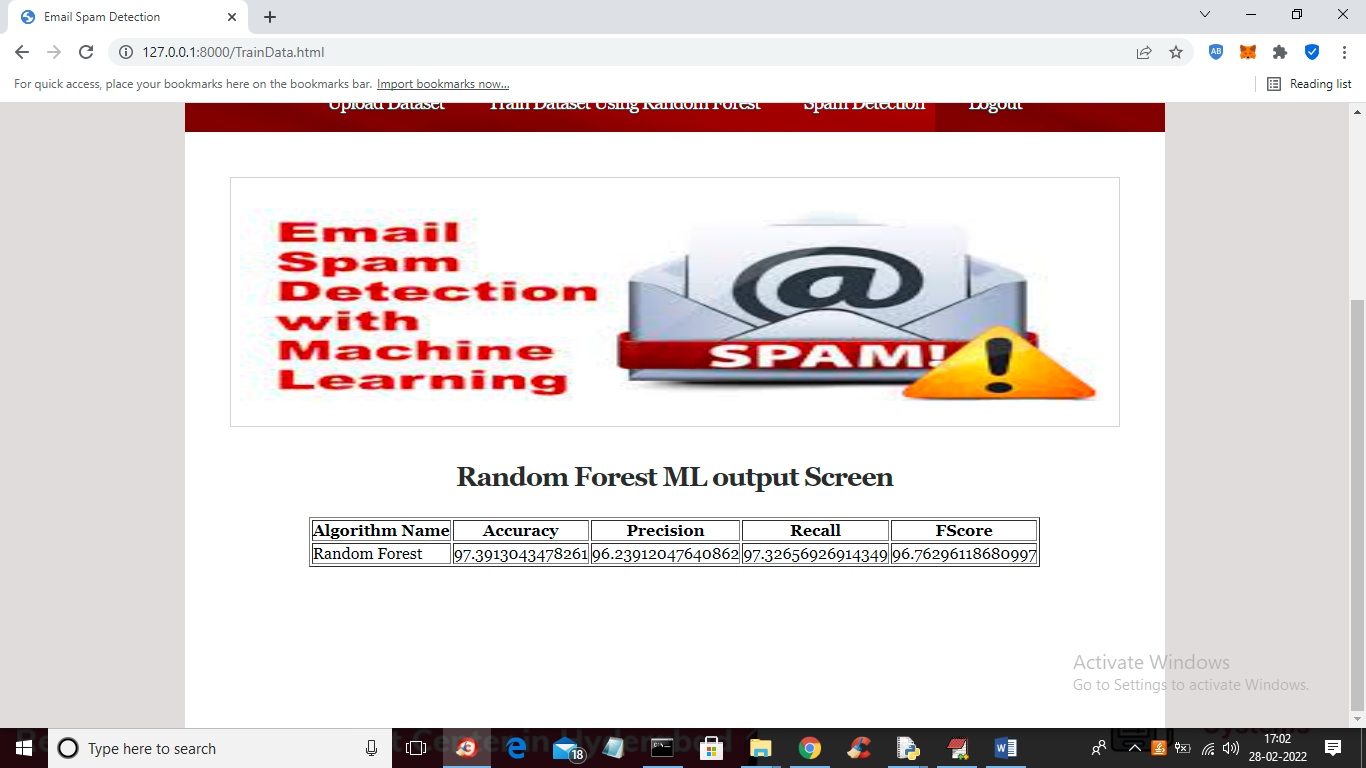
Screenshot 6.6 : Uploading Dataset

Select the particular Dataset and upload it



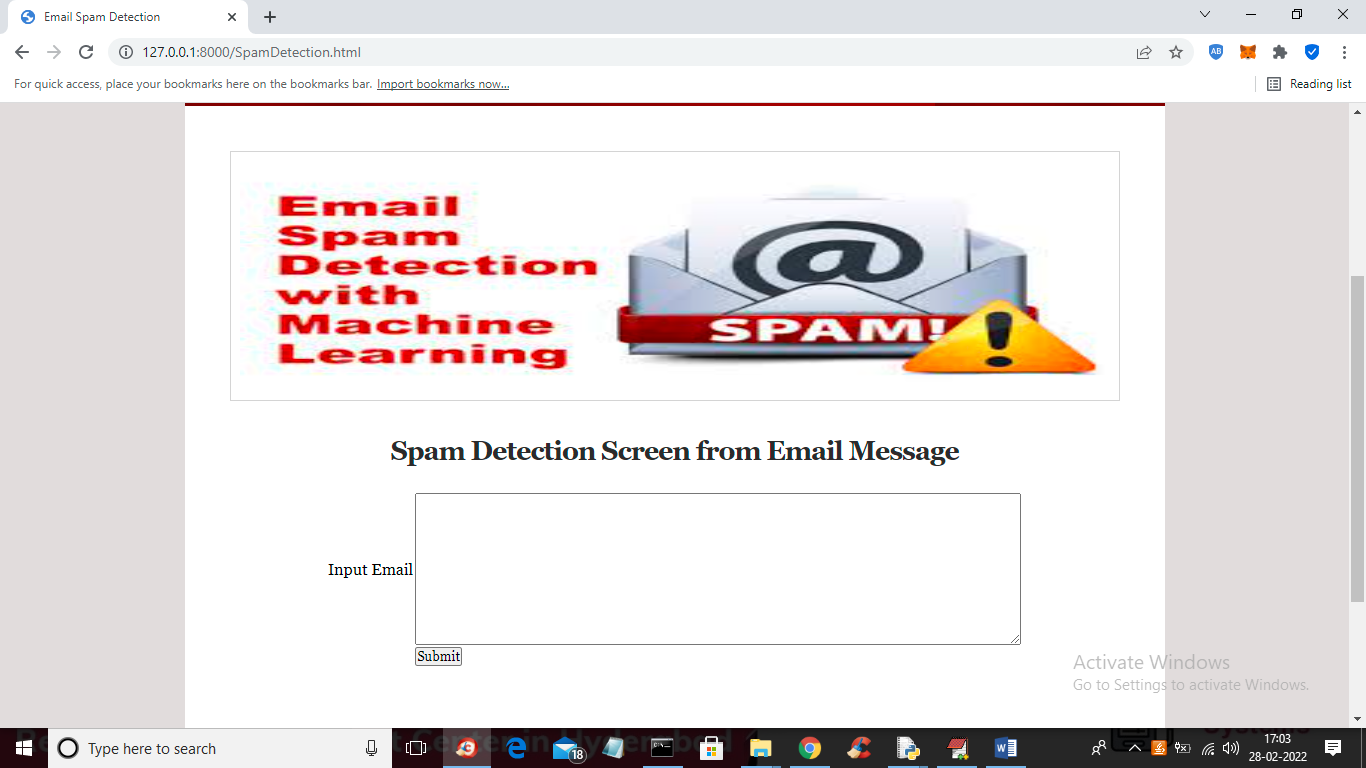
Screenshot 6.7 : Class Labels and Email Messages of Uploaded Dataset

The dataset loaded and we can see class label and email messages and now click on ‘Train Dataset Using Random Forest’ link to train random forest and get below output



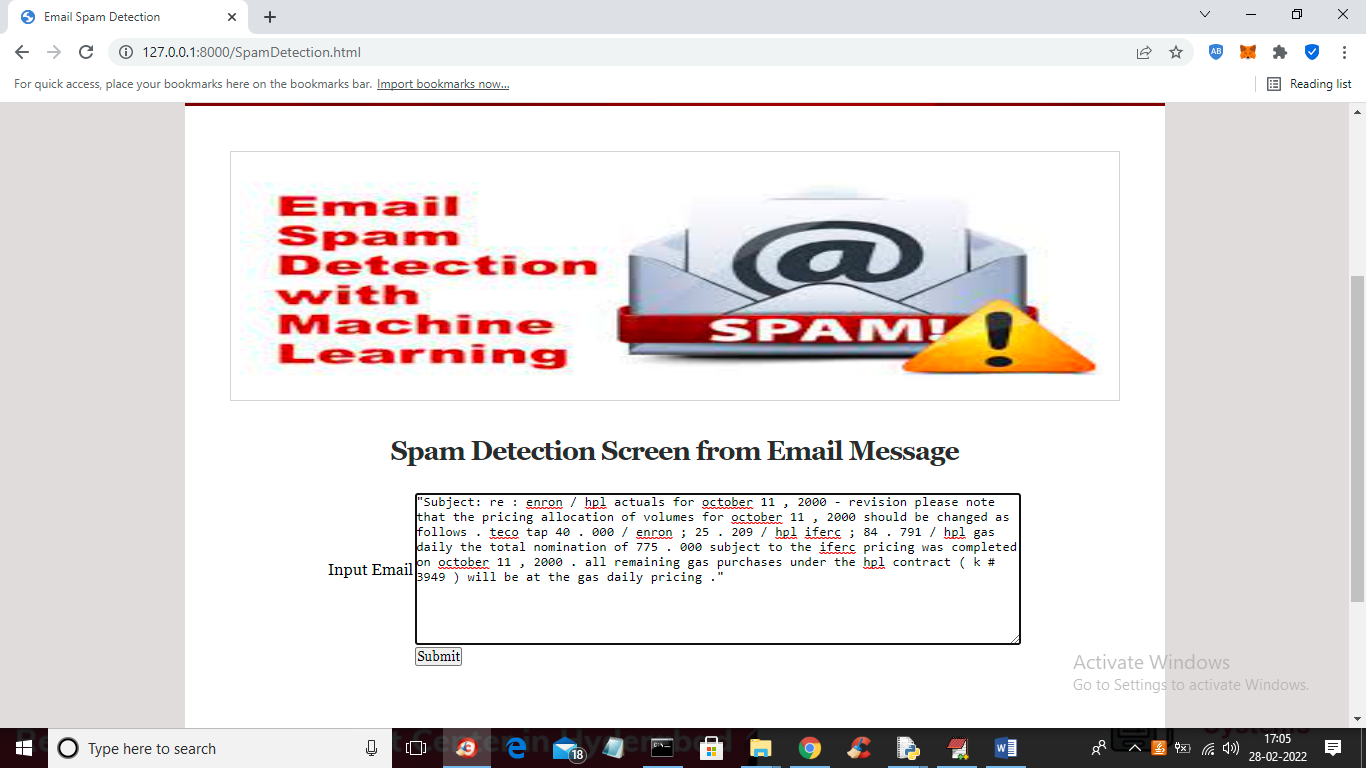
Screenshot 6.8 : Random Forest Output Screen

Random Forest trained and we got its prediction accuracy as 97% and we can see precision, recall and FSCORE. Now random forest is trained and now click on ‘Spam Detection’ link to get below screen



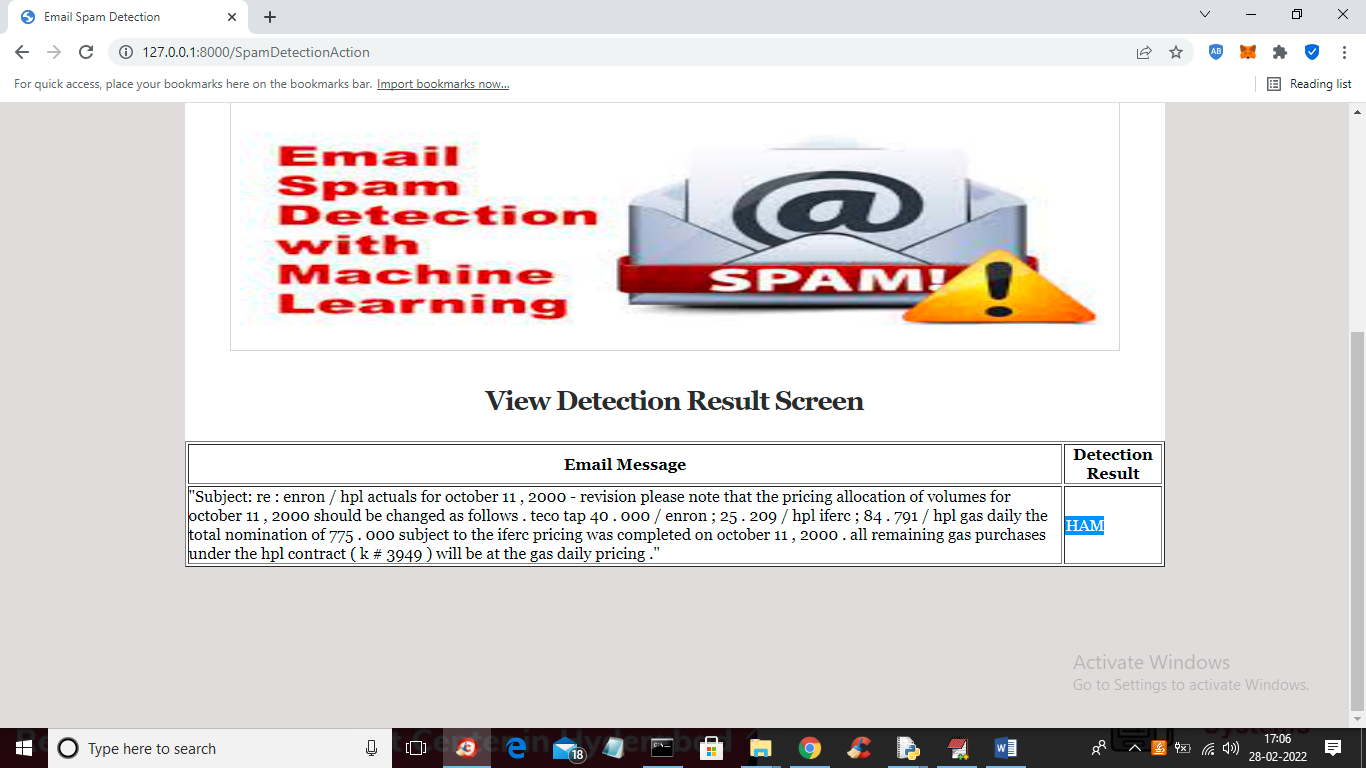
Screenshot 6.9 : Spam Detection Screen

Input some message in the above screen



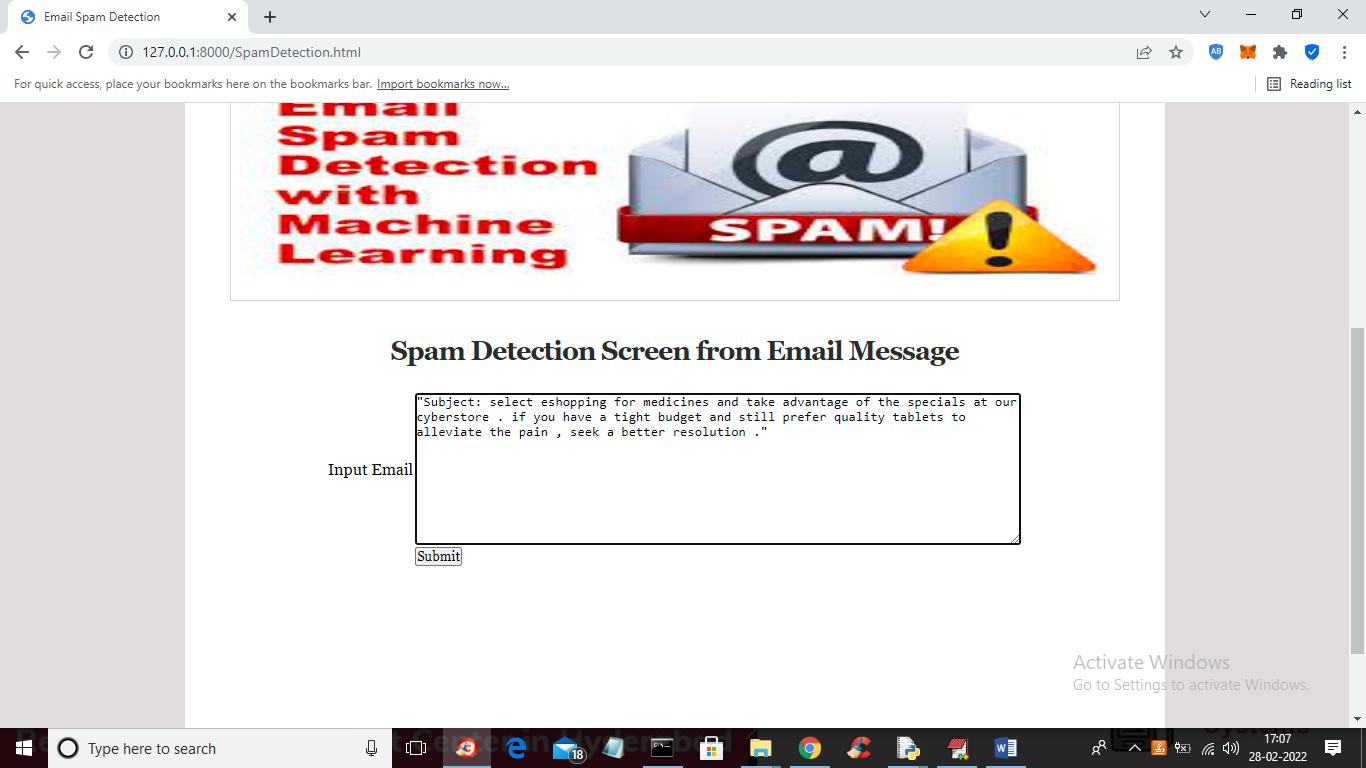
Screenshot 6.10 : Input Message

Press Submit Button to view the result of the input message



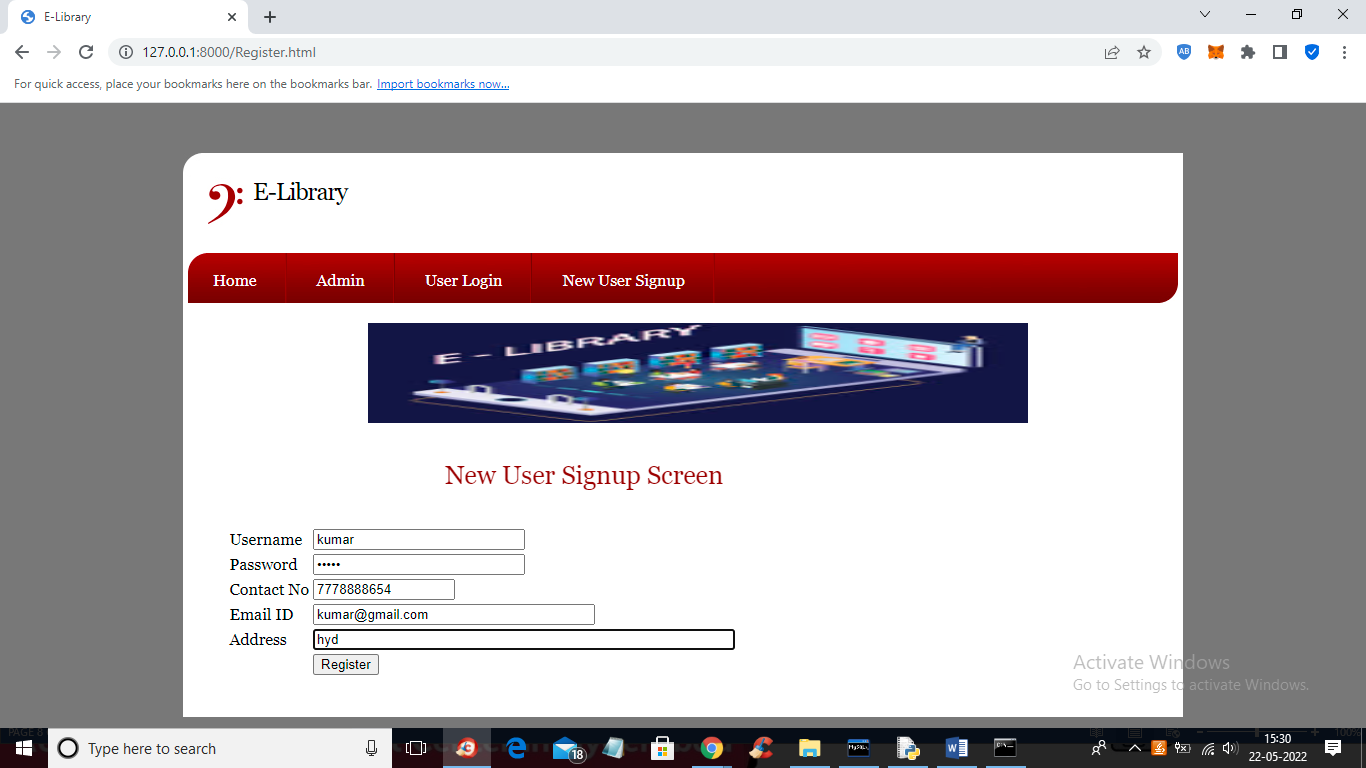
Screenshot 6.11 : Detection Result Screen

The result of the message is detected as HAM and similarly you can paste other messages and get result



Screenshot 6.12 : Other Input Message

Press Submit Button to view the result of the input message



Screenshot 6.13 : Detection Result Screen

The result of the message is detected as SPAM

## 7. TESTING

#### 7. TESTING

##### 7.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

##### 7.2 TYPES OF TESTING

###### 7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

###### 7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

###### 7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.2.4 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.2.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.2.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

# Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.2.7 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

##### 

**7.3 TEST CASES:**

**7.3.1 CLASSIFICATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **Test Case** | **Excepted Result** | **Result** | **Remarks(IF Fails)** |
| 1. | User Register | If User registration successfully. | Pass | If already user email exist then it fails. |
| 2. | User Login | If Username and password is correct then it will getting valid page. | Pass | Un Register Users will not logged in. |
| 3. | User View User | Show our dataset | Pass | If Data set Not Available fail. |
| 4. | View Fast History Results | The Four Alarm Score Should be Displayed. | Pass | The Four Alarm Score Not Displaying fail |
| 5. | User Prediction | Display Review with true results | Pass | Results not True Fail |
| 6. | Show Detection  process | Display Detection  process | Pass | Results Not True Fail |
| 7. | Show Eye Blink Process | Display Eye Blink Process | Pass | If Results not Displayed Fail. |
| 8. | Admin login | Admin can login with his login credential. If success he get his home page | Pass | Invalid login details will not allowed here |
| 9. | Admin can activate the register users | Admin can activate the register user id | Pass | If user id not found then it won’t login |
| 10. | Results | For our Four models the accuracy and F1 Score | Pass | If Accuracy And F1 Score Not Displayed fail |

**8. CONCLUSION**

##### 8 . CONCLUSION & FUTURE SCOPE

##### PROJECT CONCLUSION

The utilization of machine learning for efficient email phishing detection represents a significant stride in enhancing Cyber Security measures. The evolving sophistication of phishing attacks necessitates advanced and adaptive solutions, and machine learning provides a robust framework for achieving this. The effectiveness of machine learning in email phishing detection lies in its ability to discern subtle anomalies that may elude traditional rule-based systems. By leveraging historical data and continuously learning from new examples, machine learning models can accurately distinguish between legitimate and malicious emails, reducing false positives and enhancing overall detection accuracy. Furthermore, the real-time nature of machine learning allows for quick response and mitigation, minimizing the potential impact of phishing attacks. Integrating machine learning into email security systems not only enhances the efficiency of threat detection but also reduces the burden on end-users, as automated systems can handle a significant portion of the analysis.

##### 

##### FUTURE SCOPE

Future work focuses on Spreading sufficient awareness to detect phishing e-mail and increasing the security of companies or institutions to their users by reducing the risk of threats using highly accurate machine learning algorithms. I hope that the algorithm will be used in real life by all segments of society, allowing them to benefit from it and raise their awareness of the dangers that individuals face in society.

Feature selection techniques need more improvement to cope with the continuous development of new techniques by the attackers over the time. Therefore, we recommend developing a new automated tool in order to extract new features from new raw emails to improve the accuracy of detecting phishing email and to cope with the expanding with attacker techniques.

### 9.BIBLIOGRAPHY

##### 9. BIBLIOGRAPHY

##### 9.1 REFERENCES

* Aleroud and L. Zhou (2018), "Phishing environments, techniques, and countermeasures: A survey," Computers & Security, vol. 68, pp. 160-196, 2017.
* I. Vayansky and S. Kumar (2018), "Phishing– challenges and solutions," Computer Fraud & Security, vol. 2018, pp. 15-20.
* E. J. Williams (2018), "Exploring susceptibility to phishing in the workplace," International Journal of Human-Computer Studies, vol.120, pp .1-13.
* Ammar Odeh, (2021), "Machine Learning Techniques for Detection of Website Phishing: A Review for Promises and Challenges," in 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0813-0818.
* A. Odeh, (2021), "PHIBOOST-a novel phishing detection model using Adaptive boosting approach," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7.
* K. L. Chiew, (2018), "A survey of phishing attacks: Their types, vectors and technical approaches," Expert Systems with Applications, vol. 106, pp. 1-20.
* M. Al-Fayoumi (2020), "Intelligent association classification technique for phishing website detection," International Arab Journal of Information Technology, vol. 17, pp. 488-496.
* Y. Kwak, (2020), "Why do users not report spear phishing emails?" Telematics and Informatics, vol. 48, p. 101343.
* G. Sonowal and K. Kuppusamy (2020), "Phi DMA–A phishing detection model with multi-filter approach," Journal of King Saud University-Computer and Information Sciences, vol. 32, pp. 99-112.
* I. Keshta and A. Odeh (2021), "Security and privacy of electronic health records: Concerns and challenges,” Egyptian In f o r m a t i c s Journal, vol. 22, pp. 177-183.
* R. Faek (2021), "Exposing Bot Attacks Using Machine Learning and Flow Level Analysis," in International Conference on Data Science, E-learning and Information Systems 2021, pp. 99-106.
* A. ODEH (2020), "Efficient Prediction of Phishing Websites Using Multilayer Perceptron (Mlp)," Journal of Theoretical and Applied Information Technology, vol. 98.
* O. K. Sahingoz (2019), "Machine learning based phishing detection from URLs," Expert Systems with Applications, vol. 117, pp. 345-357.
* A. Oest (2018), "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in 2018 APWG Symposium on Electronic Crime Research (eCrime), pp. 1-12.
* A. Abbasi, et al. (2021), "The phishing funnel model: A design artifact to predict user susceptibility to phishing websites," Information Systems Research, vol. 32, pp. 410- 436.
* P. Yang (2020), "Phishing website detection based on multidimensional features driven by deep learning," IEEE Access, vol. 7, pp. 15196-15209
* Machine Learning Random Forest Algorithm - Javatpoint. (2021).Www.Javatpoint.Com. Retrieved February 17, 2022, from <https://www.javatpoint.com/machine-learning> randomforest-algorithm
* Tan, Choon Lin (2018), “Phishing Dataset for Machine Learning: Feature Evaluation”, Mendeley Data, V1, doi: 10.17632/h3cgnj8hft.1
* D. A. Pisner and D. M. Schnyer, "Support vector machine," in Machine Learning, ed: Elsevier, 2020, pp. 101- 121.
* D. Berrar, "Bayes’ theorem and naive Bayes classifier," Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands, pp. 403- 412, 2018.
* S. Athey, et al., "Generalized random forests," The Annals of Statistics, vol. 47, pp. 1148-1178, 2019

##### 9.2 GITHUB LINK

* https://github.com/Anu-Lingampalli/Email-Phishing-Detection

42