# IDENTIFICATION OF HANDLOOM CLOTHS USING IMAGE CLASSIFICATION ALGORITHM

THESIS

*Submitted by*

**A.ARAVIND KUMAR(RCAS2021MCS218)**

*in partial fulfillment for the award of the degree of*

**MASTER OF SCIENCE**
**SPECIALIZATION IN**
**INFORMATION SECURITY AND CYBER FORENSICS**


**DEPARTMENT OF COMPUTER SCIENCE**

**RATHINAM COLLEGE OF ARTS AND SCIENCE**

**(AUTONOMOUS)**

COIMBATORE - 641021 (INDIA)

**MAY-2023**

**RATHINAM COLLEGE OF ARTS AND SCIENCE**
**(AUTONOMOUS)**
COIMBATORE - 641021

# BONAFIDE CERTIFICATE

This is to certify that the THESIS entitled **DEVELOPING ENHANCED FILTERING MECHANISM TO DETECT SPOOFED MAIL** submitted by **A.ARAVIND KUMAR,** for the award of the Degree of Master in Computer Science specialization in **"INFORMATION SECURITY AND CYBER FORENSICS"** is a bonafide record of the work carried out by him under my guidance and supervision at Rathinam College of Arts and Science, Coimbatore

**Mr.S.VIGNESH**
Supervisor

**Dr.P.Sivaprakash**
Mentor

Submitted for the University Examination held on 09.05.2023

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# RATHINAM COLLEGE OF ARTS AND SCIENCE

**(AUTONOMOUS)**

COIMBATORE - 641021

## DECLARATION

I, **ARAVIND KUMAR A**, hereby declare that this THESIS **"DEVELOPING ENHANCED FILTERING MECHANISM TO DETECT SPOOFED MAIL",** is the record of the original work done by me under the guidance of **Mr.S.vignesh**, Faculty Rathinam college of arts and science, Coimbatore. To the best of my knowledge, this work has not formed the basis for the award of any degree similar award to any candidate in any University.

Signature of the Student:

ARAVIND KUMAR A

**Place: Coimbatore**

**Date: 09.05.2023**

COUNTERSIGNED

Mr.S.vignesh

Supervisor

# Contents

# Acknowledgement

On successful completion for project look back to thank who made in possible. First and foremost, thank **"THE ALMIGHTY"** for this blessing on us without which I could have not successfully our project.I am extremely grateful to **Dr.Madan.A. Sendhil, M.S.,** Chairman, Rathinam Group of Institutions, Coimbatore and **Dr. R.Manickam MCA., M.Phil., Ph.D.,** Secretary, Rathinam Group of Institutions, Coimbatore for giving me opportunity to study in this college. I am extremely grateful to **Dr.S Balasubramanian,M.Sc.,PhD,(Swiss),PDF(SwissuSA)** Principal Rathinam College of Arts and Science(Autonomous), Coimbatore.Extend deep sense of valuation to **Mr.A.Uthiramoorthy, M.C.A., M.Phil., (Ph.D),** Rathinam College of Arts and Science (Autonomous) who has permitted to undergo the project.

Unequally I thank **Dr.P.Sivaprakash, M.Tech., Ph.D.,** Mentor and **Dr.Mohamed Mallick, M.E., Ph.D.,** Project Coordinator, and all the Faculty members of the Department - iNurture Education Solution pvt ltd for their constructive suggestions, advice during the course of study.We convey special thanks, to the supervisor **Mr.S.vignesh,** who offered their inestimable support, guidance, valuable suggestion, motivations, helps given for the completion of the project.

I dedicated sincere respect to my parents for their moral motivation in completing the project.

# List of Figures

# ABSTRACT

Spam emails is a major issue on today's Internet, resulting in financial losses for businesses and annoyance for individual users. Email spoofing is a type of cyber attack in which a hacker sends an email that has been manipulated to appear to have come from a trusted source. The attacker creates a spam email attached with minor corrections in the trusted mail, which aims to bypassing the anti-spam deduction in GMAIL. This project was created with a new filtering mechanism that helps to filter the received mail to identify that mail is harmful or not. It helps to prevent email phishing attack. In this paper, The filtering mechanism is used to identify the IP-ADDRESS of the domain, Then find the exact server name of the received MAIL-ID.

# Chapter 1

# INTRODUCTION

Phishing is a type of online scam in which criminals pose as legitimate organisations through email, text message, advertisement, or other means in order to steal sensitive information. This is usually accomplished by including a link that appears to take you to the company's website to fill out your information - but the website is a clever forgery, and the information you provide goes directly to the scammers. Email phishing, also known as "deception phishing," is one of the most well-known attack types. Malicious actors send emails to users impersonating a well-known brand, then use social engineering tactics to create a false sense of urgency, leading them to click on a link or download an asset. Usually, the links lead to unsafe websites where users' devices are infected with malware or passwords are stolen. Malicious content is stored in the downloads, which are often PDFs, and when the user opens the document, the malware is installed.

## 1.1 OBJECTIVE OF THE PROJECT

The purpose of this paper is to provide a comprehensive overview of phishing email attacks and their solutions. Current security measures are ineffective at preventing phishing attacks, particularly zero-day attacks. The following are summaries and comments on the various protection measures investigated in the literature survey.

• Network-level security: It must be updated on a regular basis based on domain and IP address blacklisting. Because it is reactive in nature, it can only be updated after a pattern of abuse has been observed for some time. However, attackers can compromise legitimate users' machines in order to prepare phishing attacks, resulting in blacklisting that prevents legitimate users from using the Website.

• Authentication techniques work on two levels. As evidenced by increasingly successful phishing attacks, user level authentication using a password as credentials can simply be broken. Domain-level authentication provided by both the sender and the recipient is a method of obtaining confidential information via fraudulent emails that appear to be legitimate. We have provided a survey of the defences against these phishing email attacks. This survey contributes to a better understanding of the phishing email problem, the current solution space, and the future scope of phishing email filtering. Approaches described in the literature still have significant limitations in terms of accuracy and performance, particularly when dealing with zero-day phishing email attacks. Most phishing email classifiers are based on supervised learning, which requires them to learn before they can be used to detect a new attack; unsupervised learning,

2 which is faster but has a low level of accuracy; or hybrid (supervised and unsupervised) learning, which is time consuming and expensive. Many algorithms have been implemented, but there is still no standard technique for preventing phishing attacks in general, or zero day phishing emails in particular. Furthermore, the majority of the work is done offline. This necessitates the completion of the data collection, data analysis, and profile-creation phases first. Offline approaches are typically "reactive." This means that if phishing email features are changed, all phases must be repeated in order. How to Identify the phishing email: The majority of people are aware of some of the key indicators of a phishing email. However, some traditional things to look for when attempting to mitigate risk include

• Legitimate information: Look for contact information or other legitimate information about the spoofed organisation, then look for things like misspellings or a sender email address with the incorrect domain.

• Malicious and benign code: Be wary of anything that attempts to fool Exchange Online Protection (EOP), such as downloads or links with misspellings.

• Shortened links: Avoid clicking on any shortened links because they are used to trick Secure Email Gateways.

• Fake brand logo: Check the message for any logos that appear real, as they may contain malicious HTML attributes.

• Little text: Ignore emails with just an image and very little text because the image could contain malicious code.

## 1.2 EXISTING SYSTEM

**LEGITIMATE MAIL HEADER:**

The received field is the most important and usually the most reliable field in the email header because it is automatically added by email servers during email transmission from the sender to the receiver. This section contains information about all of the servers or computers that the email passed through on its way from the sender to the destination. An email header contains numerous Received: fields. Fields are analysed in ascending order from bottom to top. The top line of the header contains information about the mail's recipient. And the Received: line at the bottom contains information about the email's sender, domain name, IP address, timestamps, and so on.

**DNS LOOPUP:**

Internet-connected devices or resources DNS lookup is the process of determining the IP address associated with a domain name, and Reverse DNS lookup is simply the opposite process, which determines the domain name corresponding to an IP address. **A Received:** field may contain the domain name as well as the IP address associated with that domain name. If a DNS Lookup is performed using a domain name from the Received field, the IP address returned should be the same as the corresponding IP address from the Received: field. Furthermore, if a Reverse DNS lookup is performed using the IP Address in the Received: field, the domain name obtained should be the same as the corresponding domain name in the Received: field.

**DOMAIN NAME CHECKING:**

Spoofing can be detected using the lines in a mail header. This is done to look for mail servers that are involved in mail transmission. The mail transfer agent at the sender transfers the mail to the mail transfer agent at the receiver. The transfer agent on the receiver's side should be in charge of transferring mail to the receiver. The mail server that receives the mail should be in charge of transferring the mail to the next server. Line receives the mail and forwards it to the receiver's server.

**MAIL SERVER MATCHING:**

Spoofing can be detected using the lines in a mail header. This is done to look for mail servers that are involved in mail transmission. The mail transfer agent at the sender transfers the mail to the mail transfer agent at the receiver. The transfer agent on the receiver's side should be in charge of transferring mail to the receiver.

**AUTHENTICATION TECHNIQUES:**

work on two levels, As evidenced by increasingly successful phishing attacks, user level authentication using a password as credentials can simply be broken. Domain-level authentication provided by both the sender and the recipient is a method of obtaining confidential information via fraudulent emails that appear to be legitimate. We have provided a mechanism of the defences against these phishing email attacks. Approaches described in the literature still have significant limitations in terms of accuracy and performance, particularly when dealing with zero-day phishing email attacks. Most phishing email classifiers are based on supervised learning, which requires them to learn before they can be used to detect a new attack; unsupervised learning, which is faster

but has a low level of accuracy; or hybrid (supervised and unsupervised) learning, which is time consuming and expensive. Many algorithms have been implemented, but still there is no standard technique for preventing phishing attacks in general, or zero-day phishing emails in particular. Furthermore, the majority of the work is done offline. This necessitates the completion of the data collection, data analysis, and profile-creation phases first. Offline approaches are typically "reactive." This means that if phishing email features are changed, all phases must be repeated in order.

**EMAIL FILTERING MECHANISM:**

Email filtering software is a type of security tool that inspects both inbound and outbound email traffic. These tools scan emails for security risks such as spam, malware, and potentially suspicious links or attachments before they reach the end user's mailbox or leave the end user's server. This prevents messages that have been identified as potentially posing security risks from reaching their intended recipients. Whether the potential security risk was intentionally or unintentionally passed along, this software helps ensure that each end user's outgoing messages adhere to organizational policy. It also aids in the prevention of the spread of spam, sensitive information, or malicious content.

**EXISTING FILTERING MECHANISM:**

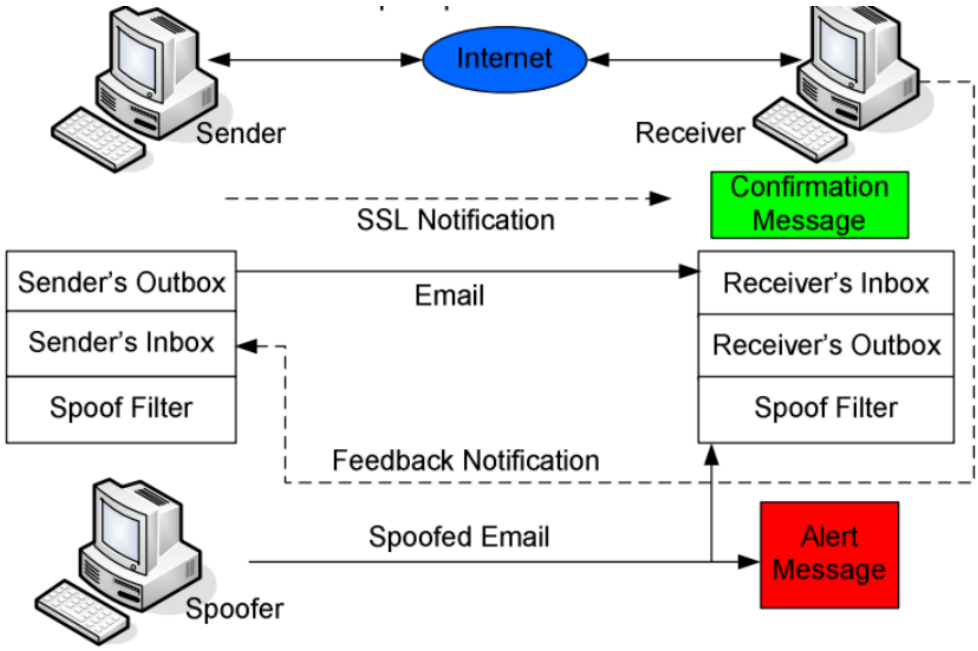



Figure 1.1: ALERT MESSAGE

# Chapter 2

# LITERATURE SURVEY

**DETECTION OF SPOOFED MAILS:2015:**

Email is important as it is one of the main forms of communication. It brings great convenience to the people, at the same time it has become a potential carrier of crime related data. Emails can be used to spread spam, viruses and other malware through spoofing. So it is important to identify spoofed mails. The proposed technique in this paper enables to detect spoofing. The major challenge in the proposed technique is that it is applicable only if email header is available for the forensic analysis.

**EMAIL APPLICATION WITH ACTIVE SPOOF MONITORING AND CONTROL:2016:**

Email spoofing is a growing security risk that necessitates sophisticated countermeasures to protect email users from its disastrous consequences. However, the majority of well-established anti-spoofing techniques are server-oriented and thus do not give users explicit control over spoof attacks.This paper extends the work of to propose a web-based client-oriented anti-spoofing application that can be deployed globally. Furthermore, the application handles alert messages and spoofed emails more efficiently, Addi-

tionally, receivers can send explicit feedback notification messages to spoofed senders. Although email is a convenient mode of communication, it has some drawbacks. Email has evolved into a potential conduit for criminal evidence. The use of email to spread pornography, fraud, and other criminal activities has Date Time. become increasingly common. Spoofing is one of the most common types of email attacks. The analysis of date and time stamps in various fields of an email header provides information about spoofing. Furthermore, the proposed technique employs DNS lookup to identify spoofed emails by utilising domain names and IP addresses present in the header. As a result, this technique provides a good solution for detecting email spoofing.

**LEGITIMATE MAIL SERVER DETECTION METHOD USING SENDER AUTHENTICATION:2021:**

This section describes the outcome of applying our new method to our dataset. Furthermore, an example of implementing an incoming mail system with sender authentication technology and an allow list (sender reputation) is provided. 1,697 domains could be reduced by excluding the domain names included in BLSP F from the extracted legitimate SPF authentication domain names. These domain names are not valid email addresses. We were able to reduce the number of IP addresses deemed to be legitimate sources by 630 using this additional procedure.Then we check to see if these reduced source IP addresses were spam sources. There was no significant difference in the ratio of emails from legitimate sources between the previous method and this method for emails judged to be non-spam, according to this result (ham).

**EMAIL SPOOFING DETECTION USING VOLATILE MEMORY FORENSICS:**

Email Forgery With the growth and evolution of email, so do the threats associated with email communication, with sophisticated adversaries employing cutting-edge techniques such as sending spoofed emails from/to legitimate user email addresses. Email spoofing is a technique that involves forging the email source in order for the email to appear to have come from somewhere else. Email spoofing is commonly used by adversaries in phishing attacks to gain the target's trust that the received email is from a legitimate source and trick them into opening the malicious email document.Our scheme, which is installed on the client machine, runs on a regular basis to determine whether the client received any spoofed emails and also detects whether the client replied to any received spoofed emails, all while preventing false positives by correctly distinguishing spoofed emails from genuine emails. The details of the experiments we conducted to demonstrate the deployment of our scheme for detecting spoofed emails are described in the following.

**A COMPREHENSIVE SURVEY FOR INTELLIGENT SPAM EMAIL DETECTION:**

The survey work presented in this paper discussed the different types of spam emails and their impact on modern society and commerce. A plethora of spam detection frameworks, both Machine Learning-based and traditional non-automated, have been critically dissected to provide a complete picture of the field's current development and future direction. It is expected that adequate development will branch out in the less

explored arena of Machine Learning-based spam identification propositions in the near future. The reviews indicate that the currently emerging frameworks, despite using automated machine learning-based solutions, are frequently unprepared to deal with the multiple angles from which an email spam threat can spread.As a result, the future direction of research in this field should be to develop anti-spam software that can combat multiple types of email spamming at the same time, taking into account the multiple angles of attack discussed above, with a single installation of the software.

**FILTERING SPAM WITH BEHAVIORAL BLACKLISTING:**

This paper presented SpamTracker, a system that classifies email senders using a technique we call behavioral blacklisting. Rather than classifying email senders according to their IP addresses, behavioral blacklisting classifies senders based on their sending patterns. Behavioral blacklisting is based on the premise that many spammers exhibit similar, stable sending patterns that can act as "fingerprints" for spamming behavior. SpamTracker clusters email senders based on the set of domains that they target. SpamTracker uses these sending patterns of confirmed spammers to build "blacklist clusters", each of which has an average vector that represents a spamming fingerprint for that cluster. SpamTracker tracks sending patterns of other senders and computes the similarity of their sending patterns to that of a known spam cluster as the basis for a "spam score". Our evaluation using email logs from an email provider that hosts over 115 independent domains shows that SpamTracker can complement existing.

# Chapter 3

# METHODOLOGY

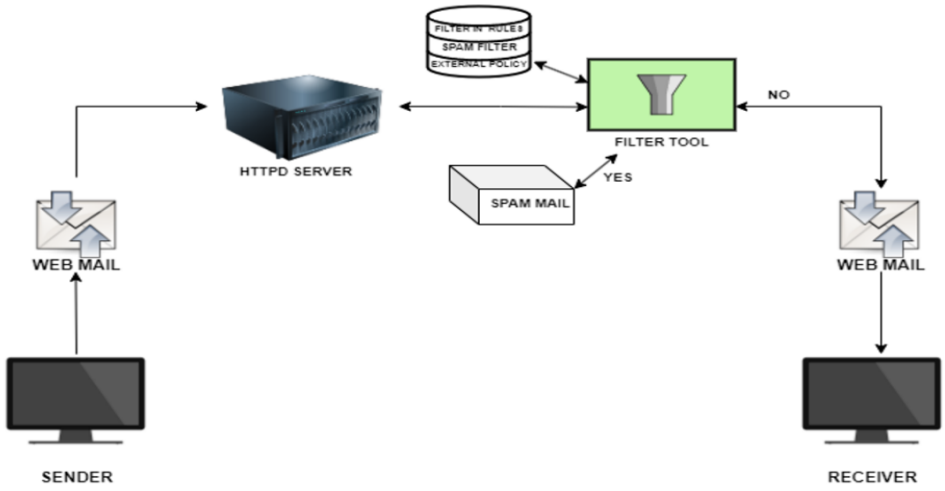



Figure 3.1: FILTERING PROCESS

**ALGORITHM FOR DETECTING FORGED EMAILS:**

Step 1: Read the message header from bottom to top line.

Step 2: Identify the various header fields to detect spoofing.

Step 3: Examine the Date: and Received: fields to determine Timestamps should be converted to UTC.

Step 4: Contrast these timestamps with the norm. A timestamp is required foremail communication.

Step 5: Use the IP address to perform a DNS or reverse DNS lookup. Available addresses and domain names in the Received: fields. .

Step 6: Cross-check the domain name in the Message-ID field. with the domain name added by in the Received: field the sender's mail server.

Step 7: Check the Received: field for mail servers that match.

Step 8: If more than two methods show an imbalance, the email is likely spoofed.

**PHASE:1**

**CREATE A OWN MAIL SERVER:**

**SOURCES INSTALLED:**

1.INSTALLED/CONFIGUR WEBMAIL/DOVECOT MAIL SERVER

2.INSTALLED EPEL-REPOSITORY

3.INSTALLED POSTFIX

4.INSTALLED SQUIRRELMAIL

**INSTALLED/CONFIGUR WEBMAIL/DOVECOT MAIL SERVER:**

**INSTALL AND CONFIGURE DOVECOT:**

yum install dovecot

gedit /etc/dovecot/dovecot.conf

Line 24 - umcomment

protocols = imap pop3 lmtp

Edit file /etc/dovecot/conf.d/10-mail.conf file

vi /etc/dovecot/conf.d/10-mail.conf

Line 24 - uncomment

mail location = maildir: /Maildir

Edit /etc/dovecot/conf.d/10-auth.conf

vi /etc/dovecot/conf.d/10-auth.conf

And make the changes as shown below:

line 10 - uncomment

disable plaintext auth = yes

Line 100 - Add the word: "login"

auth mechanisms = plain login

Edit file /etc/dovecot/conf.d/10-master.conf,

vi /etc/dovecot/conf.d/10-master.conf

**Make changes as shown below:**

Line 91, 92 - Uncomment and add "postfix"

mode = 0600

user = postfix

group = postfix

[...]

**START DOVECOT SERVICE:**

systemctl enable dovecot

systemctl start dovecot

Testig Dovecot

telnet localhost pop3

user anand

pass anand

retr 1

. quit

**INSTALLED EPEL-REPOSITORY:**

The EPEL repository is an additional package repository that provides quick access to commonly used software installation packages. Fedora contributors wanted to use Fedora packages they maintain on RHEL and other compatible distributions, so this repo was created.Simply put, the goal of this repository was to improve access to software on Enterprise Linux compatible distributions.

Install EPEL Repository:Extra Packages for Enterprise Linux (EPEL)

yum install epel-release

Allow the Apache default port 80 through your firewall/router:

firewall-cmd –permanent –add-port=80/tcp

Restart firewall using command:

firewall-cmd –reload

Restart your server to take effect all changes.

**INSTALLED POSTFIX:**

Postfix is a well-known Mail Transfer Agent (MTA) that is used to determine routes and send emails. This cross-platform server is open-source, free, and can be installed on most UNIX-like operating systems. Postfix is made up of numerous client and server

programmes: the latter usually runs in the background, while the former is used by user or administrator programmes. Postfix's structure is modular: it is made up of several small, self-contained executables. There are also various parameters, features, and options available. Another important aspect of Postfix is that it was designed to address the shortcomings of Sendmail. A strong configuration protects Postfix user data from leakage, abuse, and spam.

**INSTALL POSTFIX:**

yum install postfix

Configuring PostFix

gedit /etc/postfix/main.cf

Line no 77 - Uncomment and set your mail server FQDN

myhostname = server1.aravind.local

Line 85 - Uncomment and Set domain name

mydomain = anand.local

Line 101 - Uncomment

myorigin = mydomain

Line 115 - Uncomment and Set ipv4

inet interfaces = all

Line 121 - Change to all

inet protocols = all

Line 166 - Comment

mydestination = myhostname, localhost.mydomain, localhost,

Line 167 - Uncomment

mydestination = myhostname, localhost. mydomain, localhost,

mydomain

Line 266 - Uncomment and add IP range

mynetworks = 192.168.1.0/24, 127.0.0.0/8

Line 421 - Uncomment

home mailbox = Maildir/

Save and exit the file.

Start/restart Postfix service now:

systemctl enable postfix

systemctl restart postfix

G. Test the Postfix Server

useradd abc

passwd abc

yum install telnet

telnet localhost smtp

**Type the commands:**

ehlo localhost

mail from:¡abc¿

rcpt to:¡aravind¿

data

welcome to Postfix Mail Server

.

quit

**INSTALLED SQUIRRELMAIL:**

Squirrel Mail is a design that tries to offer an IMAP proxy server as well as a WEBBASED EMAIL CLIENT. Squirrel Mail is compatible with any other operating systemsthat support PHP and can be used in collaboration with a LAMP stack. To transmit emails, the web server needs connection to both an SMTP server and the IMAP server containing the email.SquirrelMail was added to server when webmail was first made available to server users more than a years back. Its appearance and feature set haven't changed much since its introduction.PHP-based webmail software called SquirrelMail follows industry standards. IMAP and SMTP protocol support is built-in pure PHP support, and all pages render in pure HTML 4.0 (without the need for JavaScript) for optimal browser compatibility. It is very simple to configure and install, and it only needs a few requirements. SquirrelMail offers every feature you might possibly need in an email client, such as robust MIME support, address books, and folder manipulation.

**CONFIGURING SQUIRRELMAIL:**

yum install squirrelmail

cd /usr/share/squirrelmail/config/

.conf.pl

Enter 1

Enter 1 and change the organization Name

Press S

Press R for previous menu

Enter 2 for Server settings

Press 1

and type: server1.local (Name of Domain)

Press 3

Press 2 for switching from Send Mail to SMTP

Type S and Press Q for quit

# Chapter 4

# EXPERIMENTAL SETUP

### EXISTING FILTERING MECHANISM:

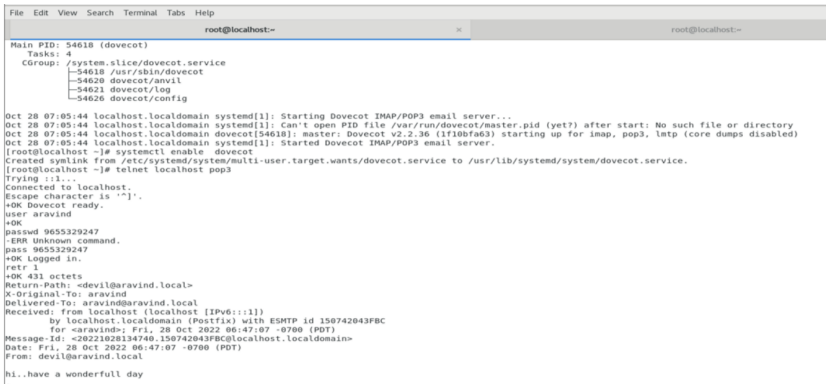



Figure 4.1: TESTING MESSAGE

### EXISTING FILTERING MECHANISM:

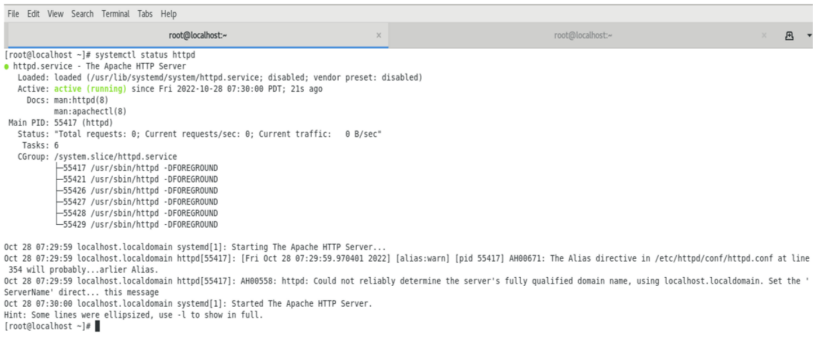



Figure 4.2: HTTP SERVER


Figure 4.3: LOGIN PAGE

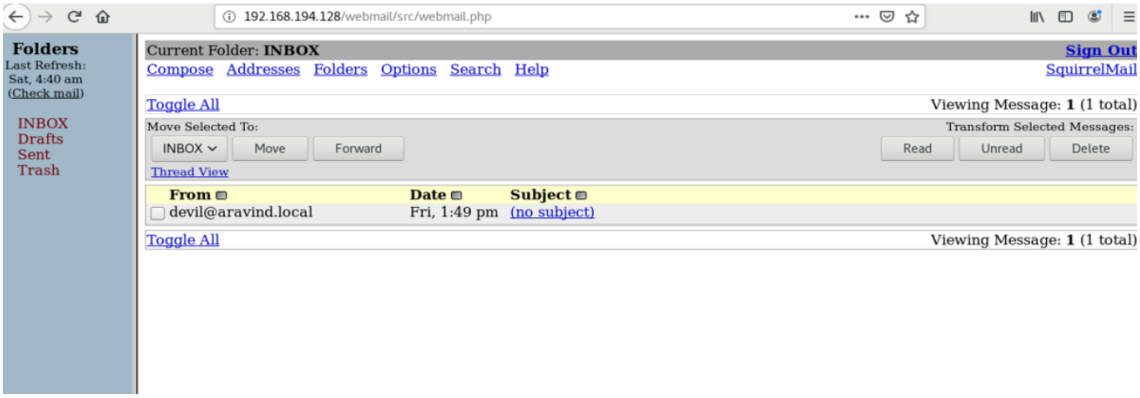



Figure 4.4: MAIN PAGE

# Chapter 5

# SPAM FILTERING MECHANISM

### MACHINE LEARNING MODEL USING NAIVE BAYES ALGORITHM

- •Data cleaning
- •EDA
- •Text Preprocessing
- •Model building
- •Deployed in stream-lit
- •Website

**DATA CLEANING:**

Data cleaning is the process of locating and eliminating or rectifying false, insufficient, or unnecessary data from a dataset. Data cleaning is a crucial stage in machine learning since the calibre of the data used to train a model can greatly affect its performance and accuracy.

Naive Bayes is a well-liked classification method that determines the likelihood of a specific result given a set of features. Because it can classify emails as spam or not rapidly based on their content, it is frequently employed in spam prediction.

## 1. Data Cleaning

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

In [7]:
```
# drop last 3 cols
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

In [8]:
```
df.sample(5)
```

Out[8]:

| | v1 | v2 |
|---|---|---|
| **1947** | ham | The battery is for mr adewale my uncle. Aka Egbon |
| **2712** | ham | Hey you still want to go for yogasana? Coz if ... |
| **4428** | ham | Hey they r not watching movie tonight so i'll ... |
| **3944** | ham | I will be gentle princess! We will make sweet ... |
| **49** | ham | U don't know how stubborn I am. I didn't even ... |

In [9]:
```
# renaming the cols
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(5)
```

Out[9]:

| | target | text |
|---|---|---|
| **1418** | ham | Lmao. Take a pic and send it to me. |
| **2338** | ham | Alright, see you in a bit |
| **88** | ham | I'm really not up to it still tonight babe |
| **3735** | ham | Hows the street where the end of library walk is? |
| **3859** | ham | Yep. I do like the pink furniture tho. |

**Exploratory Data Analysis (EDA):**

•The purpose of exploratory data analysis (EDA) is to get a knowledge of the dataset, find patterns, trends, and anomalies, and prepare the data for modelling. EDA is commonly carried out by visualising and summarising data using statistical approaches.

•To perform EDA with Naive Bayes for spam prediction, we first need to gather and preprocess the data. This may include tasks such as cleaning the text, removing stop words, and converting the text into numerical features using techniques such as Bag-of-Words or Term Frequency-Inverse Document Frequency (TF-IDF).

•Once the data is prepared, we can begin with EDA by exploring the distribution of the features in the dataset. For example, we might plot the distribution of the frequency of each word in the spam and non-spam messages. This can give us an idea of the features that are most informative for classification.

•Once the data is prepared, we can begin with EDA by exploring the distribution of the features in the dataset. For This can give us an idea of the features that are most informative for classification. Following that, we may use Naive Bayes to model the data and evaluate its performance using measures such as accuracy, precision, and recall. To optimise the model's hyperparameters, we can additionally employ techniques like cross-validation and grid search.

•Throughout the EDA process, it is critical to remember the limits of Naive Bayes and the assumptions it makes about the data.when there are dependencies between features or when the dataset is skewed, Naive Bayes may not perform effectively.

## 2.EDA

In [29]:
```
df.head()
```

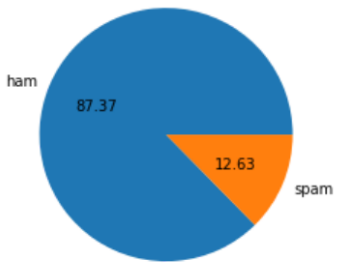
Out[29]:

| | target | text |
|---|---|---|
| **0** | 0 | Go until jurong point, crazy.. Available only ... |
| **1** | 0 | Ok lar... Joking wif u oni... |
| **2** | 1 | Free entry in 2 a wkly comp to win FA Cup fina... |
| **3** | 0 | U dun say so early hor... U c already then say... |
| **4** | 0 | Nah I don't think he goes to usf, he lives aro... |

In [31]:
```
df['target'].value_counts()
```

Out[31]:
```
0    4516
1     653
Name: target, dtype: int64
```

In [33]:
```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```




In [34]:
```
# Data is imbalanced
```

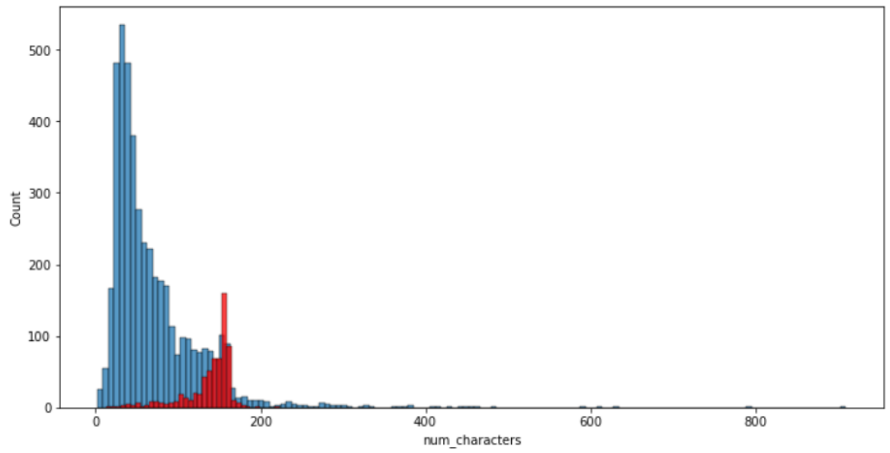
In [35]:
```
import nltk
```

In [ ]:
```
!pip install nltk
```

In [78]:
```
import seaborn as sns
```

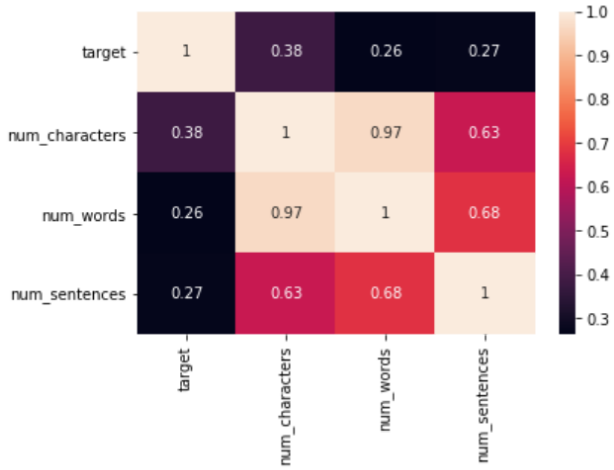
In [84]:
```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

Out[84]:
```
<AxesSubplot:xlabel='num_characters', ylabel='Count'>
```




In [89]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[89]:
```
<AxesSubplot:>
```



**DATA PREPROCESSING:**

Data preprocessing is an important stage in machine learning that involves cleaning and transforming raw data into an analysis-ready state. Here are some common data preparation techniques:

•Tokenization: is the process of separating text into separate words or tokens. Tokenization is vital for text analysis since it allows us to count the frequency of particular terms and develop models on that basis.

•Removing special characters: entails removing any characters that are not letters or numbers, such as punctuation marks or special symbols. This procedure can aid with the simplification of the data and the removal of any noise that may interfere with the analysis.

•Remove stop words and punctuation: Stop words are common words that have little significance, such as "the," "and," or "a." Removing them can help to reduce data noise and focus on more relevant words.

•Stemming: the process of reducing words to their base form, or stem. This can help to group related words together and decrease the data's complexity. For example, "jump", "jumped", and "jumping" would all be stemmed to "jump".

The Naive Bayes algorithm is a prominent machine learning technique for classification applications. It operates by assessing the likelihood of a specific input belonging to each class and then selecting the class with the highest probability as the output.

## 3. Data Preprocessing

- Lower case

- Tokenization
- Removing special characters
- Removing stop words and punctuation
- Stemming

In [187…

```
def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))


    return " ".join(y)
```

In [192…

```
transform_text("I'm gonna be home soon and i don't want to talk about this stuff any
```

Out[192…

```
'gon na home soon want talk stuff anymor tonight k cri enough today'
```

In [191…

```
df['text'][10]
```

Out[191…

```
"I'm gonna be home soon and i don't want to talk about this stuff anymore tonight,
k? I've cried enough today."
```

In [186…

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
```

Out[186…

```
'love'
```

In [194…

```
df['transformed_text'] = df['text'].apply(transform_text)
```

In [195…

```
df.head()
```

Out[195…

| | target | text | num_characters | num_words | num_sentences | transformed_text |
|---|---|---|---|---|---|---|
| **0** | 0 | Go until jurong point, crazy.. Available only ... | 111 | 24 | 2 | go jurong point crazi avail bugi n great world... |

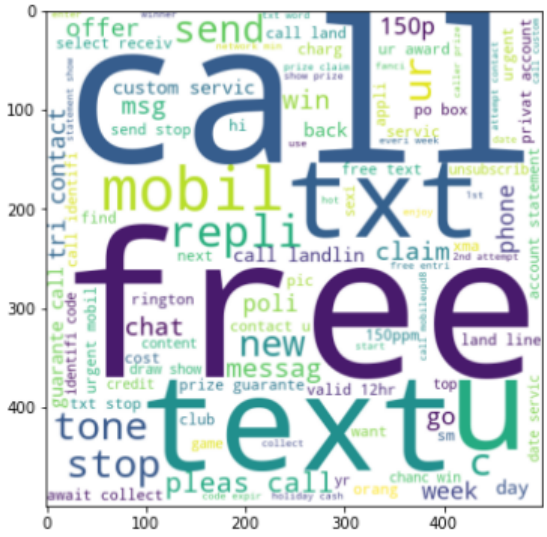
| | target | text | num_characters | num_words | num_sentences | transformed_text |
|---|---|---|---|---|---|---|
| 1 | 0 | Ok lar... Joking wif u oni... | 29 | 8 | 2 | ok lar joke wif u oni |
| 2 | 1 | Free entry in 2 a wkly comp to win FA Cup fina... | 155 | 37 | 2 | free entri 2 wkli comp win fa cup final tkt 21... |
| 3 | 0 | U dun say so early hor... U c already then say... | 49 | 13 | 1 | u dun say earli hor u c alreadi say |
| 4 | 0 | Nah I don't think he goes to usf, he lives aro... | 61 | 15 | 1 | nah think goe usf live around though |

In [232...
```
from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

In [233...
```
spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))
```

In [236...
```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

Out[236...
```
<matplotlib.image.AxesImage at 0x16f87ea8cd0>
```



**MODEL BUILDING:**

In machine learning, model building refers to the process of training a predictive algorithm or model with a dataset to make predictions or choices about fresh, unknown data.

•Naive Bayes is a probabilistic classification technique that employs Bayes' theorem to determine the likelihood of a given instance belonging to a specific class. It makes the strong and often unreasonable assumption that the features in the dataset are independent of each other, although it performs effectively in many actual applications.

•Spam prediction is one of the most common applications of Naive Bayes. In this case, the algorithm is trained on a dataset of emails, each of which is labelled as spam or not spam (ham). The algorithm learns the relationship between the words and phrases in the emails and their accompanying labels, and then utilises that information to predict whether or not a new email is spam.

•The training procedure is estimating the possibility of each word or phrase appearing in a spam or ham email and then using these probabilities to develop a model that can predict whether or not an email is spam.When a new email is received, the algorithm calculates the probability of it being spam or not based on the words and phrases it contains, and classifies it accordingly.

## 4. Model Building

In [522…
```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

In [523…
```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

In [470…
```
#from sklearn.preprocessing import MinMaxScaler
#scaler = MinMaxScaler()
#X = scaler.fit_transform(X)
```

In [483…
```
# appending the num_character col to X
#X = np.hstack((X,df['num_characters'].values.reshape(-1,1)))
```

In [524…
```
X.shape
```

Out[524…
```
(5169, 3000)
```

In [525…
```
y = df['target'].values
```

In [526…
```
from sklearn.model_selection import train_test_split
```

In [527…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [528…
```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [489…

```
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
```

In [490…

```
gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```

```
0.8916827852998066
[[808  88]
 [ 24 114]]
0.5643564356435643
```

In [529…

```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.971953578336557
[[896   0]
 [ 29 109]]
1.0
```

In [492…

```
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9835589941972921
[[895   1]
 [ 16 122]]
0.991869918699187
```

In [493…

```
# tfidf --> MNB
```

Figure 5.1: ACCURACY

**STREAM-LIT:**

Streamlit is an open-source Python framework used for building web applications. It provides an easy-to-use interface for data scientists and machine learning engineers to create interactive dashboards and visualizations.

To deploy a machine learning model using Streamlit, one can follow the following steps:

•Train a machine learning model on a dataset

•Save the trained model

•Write a Streamlit application to load the saved model and make predictions

•Deploy the Streamlit application on a web server or a cloud platform

Now, let's use the example of spam prediction using Naive Bayes to explain how Streamlit can be used to deploy a machine learning model.

Once the model is trained, we can save it using a Python library such as joblib or pickle. Then, we can write a Streamlit application that loads the saved model and accepts user input in the form of an email message. The application will then use the loaded model to predict whether the input email is spam or not.

Overall, Streamlit is a powerful tool for deploying machine learning models, and Naive Bayes is a popular algorithm for text classification tasks such as spam prediction. By combining these two technologies, we can create a powerful and interactive web application for predicting spam emails.

## STREAM-LIT WEB APPLICATION

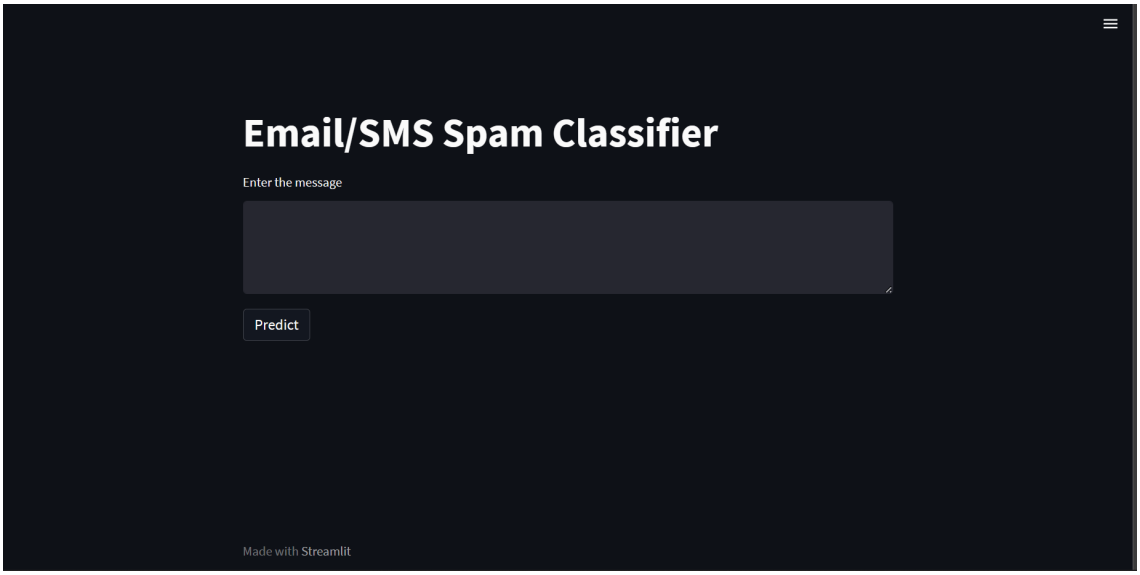



Figure 5.2: EMAIL/SMS CLASSIFIER

## OUTPUT:

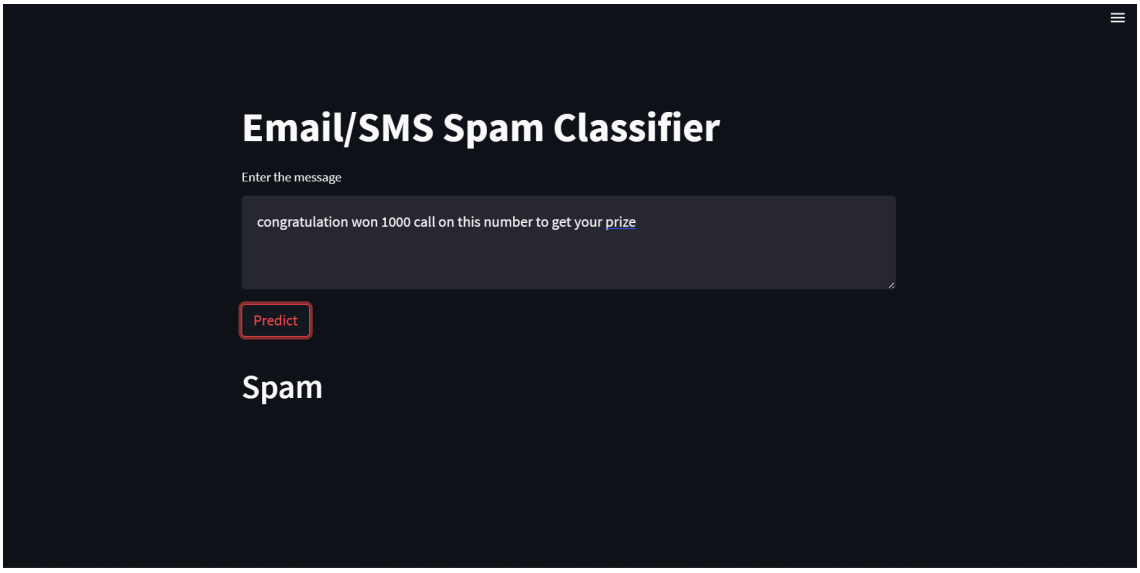



Figure 5.3: SPAM PROVED

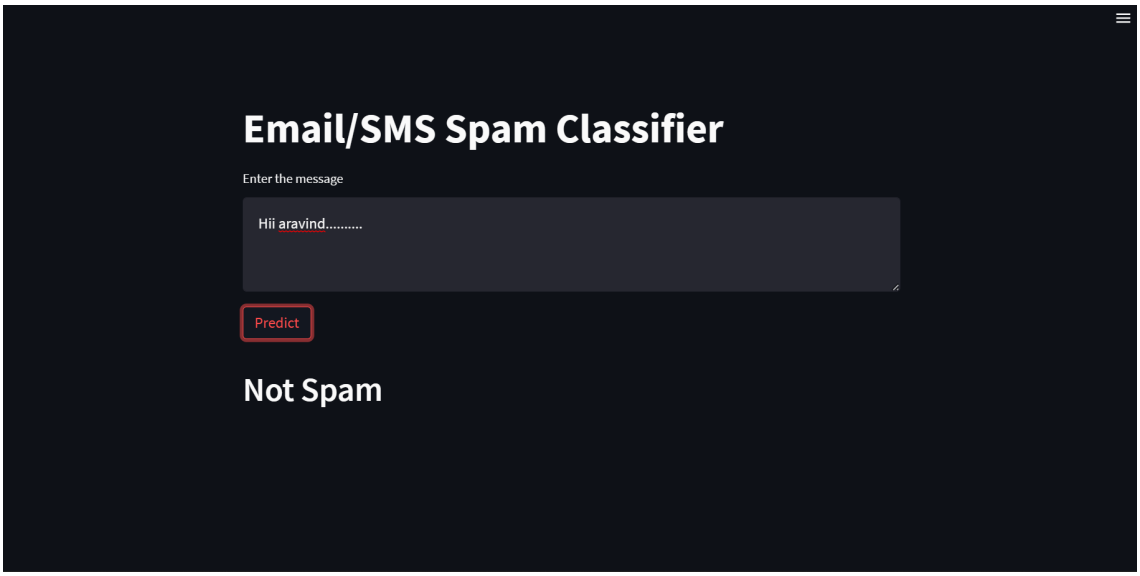



Figure 5.4: NOT A SPAM PROVED

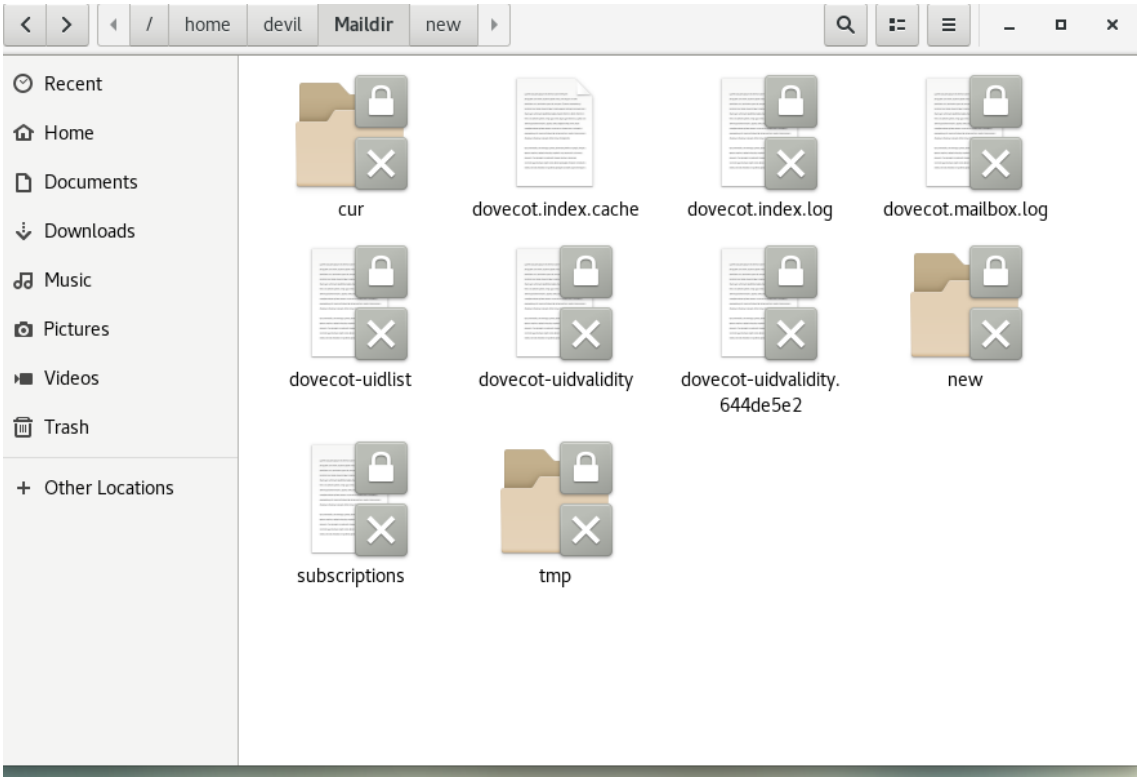



Figure 5.5: MESSAGE QUARANTINED

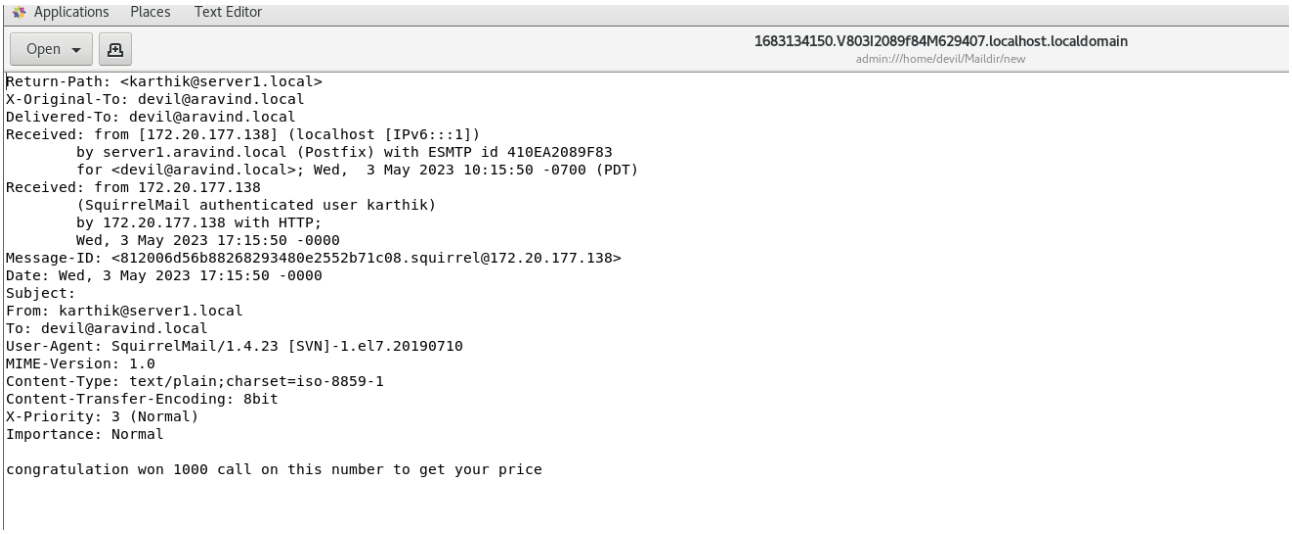



Figure 5.6: SPAM MESSAGE DETAIL IN TEXT

# Chapter 6

# CONCLUSION

This web-based client-oriented anti-spoofing application is designed to provide email users with more control over detecting and handling spoofed emails. It offers an easy-to-use interface that enables users to identify and report suspicious emails quickly. The application uses a variety of techniques to detect spoofed emails, including sender verification, domain authentication, and message integrity checks.One of the key advantages of this application is that it can be deployed globally, making it accessible to users in different parts of the world. It is also designed to be scalable, which means it can handle large volumes of email traffic without compromising its performance. This is particularly important given the increasing volume of email traffic and the growing threat of email spoofing.

Finally, the application allows receivers to send explicit feedback notification messages to spoofed senders. This can help to raise awareness about the dangers of email spoofing and encourage senders to take steps to improve the security of their email accounts.

# REFERENCES


1. A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti. and M. Alazab: "A comprehensive survey for intelligent spam email detection," IEEE Access, Vol. 7, pp. 168261-168295, 2019.

2. A. Ramachandran, N. Feamster, and S. Vempala: "Filtering spam with behavioral blacklisting," Proceedings of the 14th ACM conference on Computer and communications security, pp. 342–351, 2007.

3. D. Crocker, T. Hansen, M. Kucherawy: "DomainKeys identified mail (DKIM) signature," STD 76, RFC6376, 2011.

4. D. Crocker,T. Hansen and P. Hallam-Baker, "DomainKeys Identified Mail (DKIM) Service Overview", 2009.

5. D. Sipahi, G. Dalk l c¸ and M. H. Ozcanhan: "Detecting spam ¨ through their sender policy framework records," Security and Communication Networks, Vol. 8, No. 18, pp. 3555–3563, 2015.

6. DNS Whitelist - Protect against false positives: https://www.dnswl.org/

7. G. Clule,"Phishing attack attempts to steal Google passwords via Red Cross

website", 2013.

8. G. Velmayil and S. Pannirselvam, "Detection and Removal of IP Spoofing Through Extended-Inter Domain Packet Filter Architecture" in Int Journal of Computer Applications, Vol. 49- No.17, July 2012.

9. H. Esquivel, A. Akella and T. Mori: "On the effectiveness of IP reputation for spam filtering," 2010 Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010), pp. 1-10, 2010.

10. Hong Guo, Bo Jin, and Wei Qian, "Analysis of Email Header for Forensics Purpose", International Conference on Communication Systems and Network Technologies, 2013.

11. J. Strickland,"10 Worst Computer Viruses of All Time", Available from: http://computer.howstuffworks.com/worstcomputerviruses.htmpage=10 [Accessed February 2015].

12. J.Mehnle, "Sender Policy Framework.Introduction", 2010. Available from: http://www.openspf.org/ [Accessed February 2013].

13. K. Konno, N. Kitagawa, S. Sakuraba, and N. Yamai: "Legitimate email forwarding server detection method by X-means clustering utilizing DMARC reports," Eleventh Internation Conference on Evolving Internet (INTERNET 2019), pp. 24-29, 2019.

14. M. Kucherawy and E. Zwicky: "Domain-based message authentication, reporting, and conformance (DMARC)," RFC7489, 2015.

15. M. Rouse, "email spoofing", Search Security, 2007.

16. M. T. Banday, "Algorithm for detection and prevention of E-mail date spoofing ", International Journal of Computer Applications, vol. 06, pp. 7-11, 2011.

17. M. T. Banday, "Analysing E-Mail Headers for Forensic Investigation", Journal of Digital Forensics Security and Law, vol. 6, pp. 49-64, 2011.

18. Microsoft Corporation,"Sender ID Framework Verification SystemAims to Reduce Spam and Increase Safety Online", 2007.

19. P. Gil,"What Is an 'Email Spoof'? Is It a Type of Phishing Attack?", Available from: http://netforbeginners.about.com/od/p/f/email-spoofphishing-attack.htm [Accessed February 2015].

20. Preeti Mishra, Emmanuel S. Pilli and R. C. Joshi "Forensic Analysis of E-mail Date and Time Spoofing", Third International Conference on Computer and Communication Technology, 2012.

21.R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem,"Towards an integrated E-mail forensic analysis framework", Digital Investigation, vol. 5, pp. 124-137, 2009.

22.Radicati Group Inc, Emails Statistics Report 2013-2017.

23. S. Kitterman: "Sender policy framework (SPF) for authorizing use of domains in email, version 1," RFC7208, 2014.

24. S. Sakuraba, M. Yoda, Y. Sei, Y. Tahara and A. Ohsuga, "Sender reputation

construction method using sender authentication technologies," IPSJ Journal, Vol.62, No.5, pp. 1173–1183, 2021.

25. S. Sakuraba: "Messaging technology," IIJ Internet Infrastructure Review (IIR), Vol. 47, pp.4-9, 2020, https://www.iij.ad.jp/en/dev/iir/pdf/iir vol47 EN.pdf

26. S. M. Abdulhamid et al., "A Review on Mobile SMS Spam Filtering Techniques", IEEE Access, 2017.

27.Trivedi, S. K., Dey, S. (2014, October), "A study of ensemble based evolutionary classifiers for detecting unsolicited emails", In Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems (pp. 46-51). ACM.

28. P. Parveen and P. G. Halse, "Spam Mail Detection using Classification", vol. 5, no. 6, pp. 347–349, 2016.

29.Jyh-Jian Sheu,Ko-Tsung Chu,Nien-Feng Li,Cheng-Chi Lee, " An efficient incremental learning mechanism for tracking concept drift in spam filtering", February 2017.

30.D. Hand, M. Heikki, and S. Padhraic, Chapter 3: Visualizing and Exploring Data, vol. 30, no. 7. 2001.

31.I. A. Lawal and S. A. Abdulkarim, "Adaptive SVM for Data Stream Classification".

32.T. Hastie, R. Tibshirani, and J. Friedman, "Overview of Supervised Learning", 2009.

33.H. L. Nguyen, Y. K. Woon, and W. K. Ng, “A survey on data stream clustering and classification”, Knowl. Inf. Syst., vol. 45, no. 3, pp. 535–569, 2015.

34.Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998,July), “A Bayesian approach to filtering junk e-mail”, In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105).

35.B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wozniak, ´ “Ensemble learning for data stream analysis: A survey”, Inf. Fusion, vol. 37, pp. 132–156, 2017.

36.Trivedi, S. K., and Dey, S. (2013), “Interplay between Probabilistic Classifiers and Boosting Algorithms for Detecting Complex Unsolicited Emails”, Journal of Advances in Computer Networks.

37.Vincent Lemaire, Christophe Salperwyck, Alexis Bondu, “A Survey on Supervised Classification on Data Streams”, Springer International Publishing Switzerland 2015.

38.Drucker, H., Wu, D., and Vapnik, “Support vector machines for spam categorization”, Neural Networks, IEEE Transactions on, 10(5), 1048- 1054.

39.Alexy Bhowmick · Shyamanta M. Hazarika, “Machine Learning for Email Spam Filtering: Review, Techniques and Trends”, 2016.

40.Ali Shafigh Aski,Navid Khalilzadeh Sourati, “Proposed efficient algorithm to filter spam using machine learning techniques”, Pacific Science Review A: Natural Science and Engineering 18 (2016), Elsevier.