

Enhancing Email Management and Filtering Through Naive Bayes Based Spam Detection : A Proposed Email Application Solution

by

Muhammad Farhan Rahman

19101514

Maisha Enam

19101179

Shadman Shahreyar

19101510

Valentina Mithylin

19101242

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
May 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



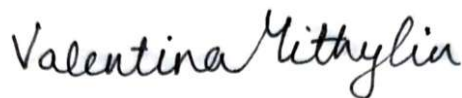
Muhammad Farhan Rahman
19101514



Maisha Enam
19101179



Shadman Shahreyar
19101510



Valentina Mithylin
19101242

Approval

The thesis/project titled “Enhancing Email Management and Filtering Through Naive Bayes Based Spam Detection : A Proposed Email Application Solution” submitted by

1. Muhammad Farhan Rahman (19101514)
2. Maisha Enam (19101179)
3. Shadman Shahreyar (19101510)
4. Valentina Mithylin (19101242)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on May 25, 2023.

Examining Committee:

Supervisor:

(Member)



Dr. Md. Khalilur Rhaman

Professor

Department of Computer Science and Engineering
Brac University

Program Coordinator:

(Member)

Dr. Md. Golam Rabiul Alam

Professor

Department of Computer Science and Engineering
Brac University

Head of Department:

(Chair)

Dr. Sadia Hamid Kazi

Chairperson and associate professor
Department of Computer Science and Engineering
Brac University

Abstract

Spam emails make up almost half of the global mail traffic. These emails take up a large amount of space in the user's inbox. However, a lot of time malware and viruses are embedded in these emails in the form of attachments or phishing links provided within the disguised emails. Moreover, sometimes important emails are flagged as spam and they are sent to spam folders causing that email to go completely unnoticed. Sometimes, even emails from educational institutions may follow a similar fate. Our goal is to build a machine learning-based email management system that not only effectively sorts and organizes emails into user-preferred categories but also contains an improved spam email detection system to ensure that important ones do not find their way into the spam folders in our inboxes. Our proposed model detects and blocks spam emails by understanding their context of it. E-mails containing newsletters, advertisements, and updates can be enhanced by adding a feature that enables machine learning to filter the email based on individual user preferences. Unwanted ones will be detected and blocked from entering the user's inbox, consequently saving space. We plan to build our proposed system using the Naive Bayes algorithm which is a computational technique employed to assess the significance of an email concerning our needs. It is a probabilistic algorithm grounded in the principles of Bayes Theorem, specifically developed to filter out spam emails for enhanced classification of the emails. An important benefit of this method for spam filtering is its adaptability to individual users, as we get more and more feedback from users it can improve its prediction. This study explores the identification of spam and non-spam emails through the utilization of the Naive Bayes algorithm. Thus, Our system learns more about the preferences of the user as time passes and can optimize its functionality accordingly. We also aim to build a web application that will make the whole process of identifying and separating emails smoother for the user.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to Sayantan Arko sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout the support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Nomenclature	x
1 Introduction	1
1.1 Problem Statement	1
1.2 Research Objectives	3
2 Background and Related Work	4
2.1 Traditional Filtering Methods	4
2.1.1 Gmail	4
2.1.2 Outlook	5
2.1.3 Yahoo Mail	6
2.2 Role of Machine Language	6
2.2.1 Supervised Machine Learning Algorithm	7
2.2.2 Unsupervised Machine Learning Algorithm	8
2.3 Data extraction	8
2.4 Types of Spam Email	8
2.5 Previous Work Related to Email Spam Detection and categorization	18
2.5.1 Related work in email categorization [3]	18
2.5.2 Existing work related to email spam detection	19
3 Initial work plan and proposed methodology	22
3.1 Preliminary Analysis	24
4 Working with Dataset	28
4.1 Dataset Description and Collection	28
4.2 Data Labeling and Naming Format	28

4.3	Data Pre-Processing	28
4.3.1	Convert to LowerCase and Punctuation	28
4.3.2	Removal of Stop Words	29
4.3.3	Stemming	29
4.3.4	Tokenization	29
4.3.5	Lemmatization	29
5	Working Model and Implementation	30
5.1	Spam Detection Using Naive Bayes Theorem	30
5.2	Web Based Application	33
6	Experimental Result	38
7	Research Contribution and Challenges	41
7.1	Contribution	41
7.1.1	Dataset Collection	41
7.1.2	Multiple AI Used	41
7.1.3	Web Application and Categorization	41
7.2	Challenges	42
7.2.1	Concern For Privacy	42
7.2.2	Automatic Deletion of Spam Mail	42
7.2.3	Third Party Authorisation	42
7.2.4	Integration of Naive Bayes and Particle Swarm Optimization for Spam Detection	42
7.3	Shortcomings	42
8	Future Work and Conclusion	46
8.1	Future Work	46
8.1.1	Better User Interface	46
8.1.2	Creating an extension	46
8.1.3	Connecting Google Scholar With The Web Application	46
8.1.4	Connecting Google Calendar With The Web Application	46
8.1.5	Detect and Filter Bengali Spam Content	47
8.1.6	Filter Social Media Spam Content	47
8.2	Conclusion	47
	Bibliography	49

List of Figures

2.1	Different Types of Spam Email	9
3.1	Initial Work flow of our system	23
3.2	F-measure of the existing models	25
3.3	FP rate of the existing models	26
3.4	Training time of the existing models	27
4.1	Stemming and Lemmatization	29
5.1	Work flow of our system	32
5.2	Starting the app	34
5.3	Getting the email	35
5.4	Entering the email	35
5.5	When it detects as Not Spam	35
5.6	Adding the email to a category	36
5.7	When it detects as Spam	36
5.8	Marking a falsely detected Spam email	36
5.9	Web based Application	37
6.1	Heatmap of the model	38
6.2	Accuracy of the model	39
6.3	Graph comparison of our model with the existing models	40
7.1	General Diagram of our Initial model	43
7.2	PSO [7]	45

List of Tables

2.1	Previous Work Related to Email Spam Detection	21
3.1	comparison between different classifiers on a test dataset. (from [6]) .	26
6.1	Performace of our model comparing with the existing models	40

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	artificial intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ART	Author-Recipient Topic
CFS	Correlation Based Feature Selection
CSS	Cascading Style Sheets
CSV	comma separated value
DT	Decision Tree
EMFET	Email Feature Extraction Tool
FN	False Negative
FP	False Positive
HTML	Hyperlink Text Marked Language
IP	Internet Protocol
KNN	K-nearest Neighbour ML algorithm
LDA	Latent Dirichlet Allocation
ML	Machine Learning
POS	Part of Speech
RART	RoleAuthorRecipient Topic
SRD	Sender Reputaion Data Network
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

UCE Unsolicited Commercial Emails

WSGI Web Server Gateway Interface

Chapter 1

Introduction

Electronic mail (email) was first introduced in 1965, and over the years, through constant improvement, it has become one of the most commonly used methods of communication. At present, email is considered to be a fast and cheap means of transferring any kind of information and electronic data. It is both easily accessible and replicable, making email a very convenient platform for business communication. However, there are still some drawbacks that decrease user experience. It is this convenient accessibility that has been exploited and used as a means of distributing scam messages and phishing links containing malicious content. Opening these links or attachments provided in the emails may result in malware and viruses intruding on the user's device. Scammers have been able to disguise these harmful contents in the form of advertisements and other types of services, so they can be easily shared by being undetectable to a regular user. In addition to that, businesses and organizations send emails in bulk to users to advertise and promote their services to gain more user participation. However, these commercial bulk messages create problems for the recipients, as their inboxes get flooded with unwanted content and news. They also take up a huge amount of memory space to be stored, thus wasting useful storage capacity and hindering efficient usage of network bandwidth as well. As time goes on email spam would become more sophisticated and more prevalent, resulting in far greater stress on not only the email servers but also the users' bandwidths. To tackle these email spam problems, renowned email service providers, like Gmail and Yahoo, use spam filters that implement machine learning algorithms for effectively filtering spam from useful messages. But, oftentimes, users also experience trouble when certain useful and important emails end up in their spam folders. We aim to enhance the existing spam filtering method and create an email management system that will use machine learning to understand the user and organize their emails based on individual preferences.

1.1 Problem Statement

Email has always been a necessary medium for enabling rapid and inexpensive communication. It has been a critical part of online communication and an integral communication method for exchanging important official information. However, spam or unsought email puts a hamper on this form of communication. Spam email floods the inbox and it makes the inbox clogged with useless emails and makes it hard to find important ones. Moreover, there is also the possibility of detecting

an important email as a false positive, otherly known as “ham”. This results in important emails not being able to reach the intended user and causing problems in communication. Shockingly, out of a vast amount of business emails, 70% of them are considered spam which leads to problems like overflowing the user’s mailbox according to the estimation of research work [6]. These vast amounts of spam email hamper productivity by not only creating traffic and using precious bandwidth but also creating problems for important emails. To solve this problem, well-known email services such as Google, yahoo came up with their anti-spam algorithms to create a better experience. But in reality, these email filters fail to stop spam emails and on the other hand, sometimes falsely identify intended emails as spam which leads to a user not noticing an important one. From a recent survey[11] it was shown that during the pandemic 61% of Employee Reported Phishing emails are False Positives. Moreover, spam email not only hampers productivity but also poses a high-security risk for users. An email-based phishing attack is still one of the most prevalent and successful ways to infect malware or get hold of private data or initiate ransomware. According to a statistical report by Kaspersky [19], spam emails reigned an average of 45.56t% of the mail traffic globally in the year 2021. However, the highest proportion of spam emails that year was in June with a rate of 48.03%. That makes up almost half of the whole global mail traffic, only with spam. In [19] found 148,173,261 attachments in the emails containing malicious content and 253,365,212 phishing links were detected and blocked, and a further 341,954 attempts to go after given phishing links were blocked. Furthermore, nowadays spam emails have been getting more and more sophisticated. Always coming up with new methods to bypass filters and innovating new ways to infiltrate, mask or pretend as legitimate email. There has been a continuous cat-mouse chase between email filters and anti-filter techniques. According to [2], to break through email filters, anti-filter techniques use genetic algorithms to create different chromosomes based on keywords in a regular expression. After training the anti-filter gets a database for later use to bypass the filter. Moreover, whenever an email filter results in a false negative, it gives the anti-filter valuable information so that their algorithm can be modified accordingly. Thus spam mail is constantly changing and innovating to stay one step further than preventative measures. With the amount of email that is sent and received daily, it is noticed that users’ inbox gets flooded with countless emails. These emails indiscriminately stay in the inbox and cause productivity issues. The usefulness of each email varies from user to user and email to email thus managing a block of the email becomes difficult and more taxing than it needs to be. As a result, a way to categorize and automatically sort emails becomes a very useful feature. Not only does it enable them to work with emails in batches but also gives ability and choice to users so that they can give attention to the necessary ones more. This also makes the inbox organized and personalized with the benefit of finding the right email whenever it is needed resulting in more efficiency. All in all, this research aims to combat email spam and stop unnecessary emails and shield users from malicious ones. Moreover, this research wants to develop a new paradigm of using email by an automatic email sorting method and improving the user experience of interacting with emails.

1.2 Research Objectives

This research aims to develop an integrated email management system with a better accuracy rate for spam detection and the opportunity for users to classify their emails into their own suitable categories. Currently, the existing spam detection algorithms have a mentionable percentage of false spam detection and, and does not allow the users to categorize email to their will. The objectives of this research are,

1. To create a better-optimized E-mail spam detection system through Naive Bayes.
2. To allow the users to create categories of their emails according to their own preferences.
3. To create a web application that can detect and notify if the mail is spam or ham
4. To decrease the percentage of fake spam detection that the existing spam detection algorithms have.

Chapter 2

Background and Related Work

As our main focus is to build an effective email communication system. Let us first look into how Gmail categorizes its emails and then how Gmail, Yahoo, and Outlook's spam filter works.

2.1 Traditional Filtering Methods

Gmail tabs utilize a grouping method based on machine learning to identify where to place information based on several signals. Signals include who sent the email, the type of things mentioned in the message, and how users of Gmail have responded to much the same information in the past. The following section explains how Gmail categorizes its emails as mentioned in [20].

Gmail categorizes messages based on several signals:

- Primary: emails from individuals you have contact with
- Social: Messages from social media networks and similar websites.
- Promotions: Consist of emails containing various deals, newsletters, and offers.
- Updates: Consists of emails like notifications about a meeting, confirmations, and receipts of payments, statements, and invoices.
- Forums: Consists of emails from a variety of online groups and discussion forums.

Users have the option of selecting one, some, or all of these groups. Gmail adapts to our choices and behaviors automatically.

The most significant input is our direct input. Gmail learns from our activities how to organize our email depending on our preferences. Here are four steps to teach Gmail to categorize our emails.

2.1.1 Gmail

Google uses a variety of methods to keep Gmail spam-free. such as IP reputation point, user engagement, the content of the email, history, etc.

1. **IP and Domain Reputation:**

Gmail takes into account the sender's domain and the IP address to know where the email should be placed. Communications can be avoided getting banned or censored by utilizing enough authentication in the outgoing email.

2. **User Engagement:**

When marking emails as spam, The algorithm of Gmail appears to give significant importance to user activities in the inbox.

The following are some instances of possible user actions:

- Messages that were deleted even before reading it.
- Messages that are categorized as spam.
- emails that were voted as non-spam
- Messages have been transferred to promotions.
- Messages with stars
- Messages passed on
- Messages received and read
- Messages returned in response to Spam reports or complaints

3. **Content:**

The header, text, graphics, and links in your email are all important variables, all of these have involvement in what category an email falls in (spam, promotional, social, or inbox). The broad filtering mechanism still includes content, but the importance seems to be comparatively dependent on the sender's internet and email history.

4. **Past Sending History:**

Gmail's normal approach for new IP addresses is to temporarily block them for the first two to twenty- four hours [20]. Based on that, a small amount of emails accumulates in the user's inbox. afterward, a small group of emails is sent to spam to see the reaction of the recipients. If this starting test comes with a substantial number of complaints, most subsequent emails will be sent to spam. Gmail would consider the address safe for inboxing if receivers unspam the spam message.

2.1.2 Outlook

Microsoft Outlook utilizes the Microsoft sender reputation data network to judge whether an email is a junk or not junk.

- **Microsoft Sender Reputation Data Network (SRD):**

The Spam Fighters program employs a group of people to vote drawn at random from Outlook users who are active to assist their filters. emails may be re-sent with a message asking panel members of SRD to vote on whether or not the main email was "Junk" or "Not Junk." A huge amount of "Junk" votes, as expected, will reduce the likelihood of your future emails reaching us. SRD can be more dependable than simply tracking complaint rates. Senders may simply influence complaint rates by sending more emails to lower the number

of complaints. It's more difficult to unnaturally decrease the complaint rate using SRD by dispatching a large number of emails.

2.1.3 Yahoo Mail

Yahoo Mail uses various data to filter out spam emails. According to Yahoo, [20] filters look for the following things:

- Internet Protocol address reputation
- Uniform Resource Locator reputation
- Domain, Sender and Autonomous System Number reputation
- Domain Keys Identified Mail signatures
- Domain Keys Identified Mail signatures

2.2 Role of Machine Language

When analyzing the email filtering system using ML algorithms, specifically spam detection, it is vital to speculate both ML and other present-day procedures that are used to detect mail as spam. Through research, it is known that the information and workings of a spam email differ over the course of time. Because of this, current procedures may become obsolete in the near future. This behavior is known as conceptual drift. Machine Learning is an engineering method developed to allow computer instruments to function without being explicitly programmed. Due to the Machine Learning system's capacity to grow as time passes, limiting concept drift, this strategy is a significant help in detecting and combating spam. In the next part, we will look into a variety of Machine Learning techniques and algorithms, as well as the benefits associated with Supervised, Unsupervised and Semi-Supervised Machine Learning algorithms, as mentioned in [17].

- **Supervised Machine Learning Algorithm:**
It is a subcategory of both Machine Learning and Artificial Intelligence. It makes use of labeled datasets for training the algorithms. Because of this, it can predict the outcome of the data. This method can be divided into two types - Regression and Classification. It is utilized to predict categorical outputs.
- **Unsupervised Machine Learning Algorithm:**
The name of the algorithm suggests that in unsupervised machine learning algorithms, the model is not supervised. The model is allowed to work on its own to discover information that may not be visible to us. It concludes with unlabeled data. It can be used for clustering, association, and dimensionality reduction.
- **Semi-Supervised Machine Learning Algorithm:**
During the testing stage of this method, the system is trained with both labeled and unlabeled data, and system analysis is performed using both techniques.

The primary goal of this method is to attain greater precision and accuracy than original supervised and unsupervised procedures. In this method, there are two sorts of output presentations: Semi-Supervised Clustering and Semi-Supervised Classification.

The next part will focus on the various ML algorithms that were used in the examined research. These were studied after being classified using the aforementioned machine learning algorithm technique.

2.2.1 Supervised Machine Learning Algorithm

- **Artificial Neural Network (ANN):**

ANN is created using artificial neurons. The system decides how many neurons are needed, and this number can be altered as required by the system. The neurons are interconnected in different layers, together with the input layer, the hidden layer, and the output layer. ANN systems 'learn' via a process known as 'back-propagation.' The network's new output is speculated and matched with the optimum match that should have been made.

- **Naive Bayes Machine Learning Algorithm:**

This is a well-known supervised machine learning algorithm. This was created using Bayes' theorem, which attempts to calculate the likelihood of an event occurring based on past information and conditions. This technique is very simple and quick to integrate into a system. The Naive Bayes algorithm considers the characteristics to be independent of one another. This system generates the intended output using a Decision Tree and Naive Bayes.

- **Support Vector Machine:**

Support Vector Machine (SVM) is a famous and widely used ML technique. Some systems only use SVM as their classification method, while others use a combination of techniques, which includes SVM. The SVM technique generates a hyperplane from which multiple classes are generated to assess various attributes collected from the dataset.

- **Decision tree (DT):**

One more algorithm that has been utilized more frequently in the studied supervised learning method research is the decision tree machine learning algorithm. The reason for using this more frequently is because it is a simple method with simple explanations and images. This method may be applied to both big and small data sets. This method can handle both numerical and categorical data.

- **Random Forest Algorithm:** Random Forest is one of the supervised machine learning algorithms that merges multiple decision trees to get a more accurate prediction, as mentioned in [16]. This algorithm helps in the early stage of developing the model when the model needs to be trained. The algorithm's ability to be simple and diverse allows it to be one of the most widely used machine learning algorithms. The Random Forest algorithm is advantageous in working with both classification and regression problems.

2.2.2 Unsupervised Machine Learning Algorithm

- **K-nearest Neighbour ML algorithm (KNN):**

K Nearest Neighbor is a Supervised ML technique that may be used to predict classification and regression concepts. KNN is a sluggish learner. Because it uses distance to classify data, normalizing the training data can enhance its accuracy significantly.

- **K- means Clustering ML algorithm:**

This technique is simple to integrate and has a lower computational cost than the KNN ML algorithm. Because of this, this method is very famous for spam detection. The data mining process starts with a group that is chosen at random. There is a centroid that is chosen randomly for each individual cluster to start with. To get the optimal solution calculations are repeated and each calculation starts from the centroid.

2.3 Data extraction

Before classifying the emails as spam or non-spam, it is very important to separate each and every feature of an email to get them checked individually, for the presence of any malicious content. We use EMFET [8] in our model to extract the features from an email, and classify them accordingly. According to the researchers [8], The features are divided into three groups using the extraction tool, which are header features (Metadata features and subject features of the email), payload features (the body, readability, and lexical diversity of the email), and attachment features (files or documents attached along with the email). A thorough analysis is carried out by this tool giving us an output file containing all the extracted features sorted out as per the three aforementioned categories.

2.4 Types of Spam Email

- **Unsolicited Commercial Emails:** Unsolicited Commercial Emails (UCE) refer to promotional or advertising messages that are sent without the recipient's prior consent. These emails are typically sent in bulk to a large number of recipients and often contain content unrelated to the recipients' interests or needs. The primary purpose of unsolicited commercial emails is to promote products, services, or offers to a wide audience, hoping to generate sales or leads. Here are some examples of unsolicited commercial emails:

- **Product Promotions:** These emails advertise various products, such as electronics, clothing, beauty products, or home appliances. They may offer discounts, limited-time offers, or exclusive deals to entice recipients to make a purchase. Example: "Get 50% off on all electronics! Limited time offer - Shop now!"
- **Service Offers:** Emails in this category promote different services, including web design, SEO optimization, financial consulting, or travel bookings. They often highlight the benefits of the service and emphasize why

Types of Spam Email

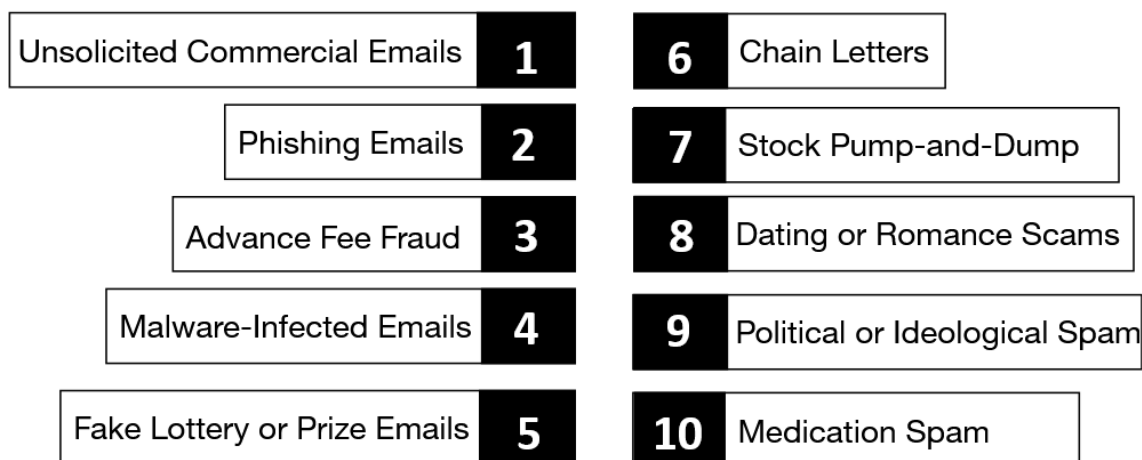


Figure 2.1: Different Types of Spam Email

recipients should choose them. Example: "Boost your website's visibility with our professional SEO services!"

- Financial Opportunities: These emails may promise investment opportunities, quick money-making schemes, or financial advice. They often make exaggerated claims about potential profits and may require recipients to provide personal information or invest money. Example: "Earn \$10,000 a week with our proven investment strategy!"
- Job Offers: Some spam emails claim to offer job opportunities or work-from-home positions. These emails may promise high-paying jobs or flexible schedules but usually aim to collect personal information or sell training programs. Example: "Work from home and earn \$5,000 per week! No experience required."
- Online Dating or Adult Content: These emails may advertise dating websites, adult content, or adult-oriented products. They often use provocative language or explicit imagery to attract recipients' attention. Example: "Find your perfect match - Join our exclusive dating site today!"
- **Phishing Emails:** Phishing emails are fraudulent messages sent by cyber-criminals, disguised as legitimate and trustworthy entities, with the intention of tricking recipients into revealing sensitive information or performing actions that may compromise their security. Phishing emails are carefully crafted to imitate reputable organizations, such as banks, online services, government agencies, or well-known companies. The emails often employ various tactics to manipulate recipients, exploiting their trust and attempting to deceive them into taking actions that benefit the attackers. The ultimate goal of these phishing emails is to acquire confidential credentials, like login data, credit card elements, or personal ID details..
 - Account Verification: An email claiming to be from a popular online service, such as a social media platform or e-commerce website, might

request that the recipient verify their account by clicking on a link and entering their login credentials. The link leads to a fake website designed to steal the entered information. Example: "URGENT: Verify Your Account Now or Risk Suspension!"

- Financial Institution Scam: An email impersonating a bank or financial institution might inform the recipient of unauthorized account activity or a security breach. The email urges them to click on a provided link to confirm their identity and secure their account. In reality, the link leads to a fraudulent website that captures the victim's banking details. Example: "Important Security Alert: Confirm Your Account Information Immediately!"
 - Tax Refund Request: A phishing email pretending to be from a tax agency or government department may inform the recipient of an unclaimed tax refund. The email prompts them to click on a link and provide personal information to process the refund. The link leads to a fake website aimed at harvesting the victim's sensitive data. Example: "Tax Refund Notification: Claim Your Unclaimed Refund Today!"
 - Urgent Payment Notice: An email posing as an invoice or payment request from a well-known vendor or service provider might urge the recipient to open an attached file or click on a link for more details. The attachment or link contains malware designed to infiltrate the victim's system or capture sensitive information. Example: "Invoice Payment Reminder: Please Review Attached Document."
 - Charity Scam: A phishing email exploiting people's goodwill may request donations for a charitable cause or a recent disaster. The email provides a link to a fake donation website where the victim is prompted to enter their credit card details, resulting in financial loss and potential identity theft. Example: "Help Victims of XYZ Disaster - Donate Now!"
- **Advance Fee Fraud:** Advance Fee Fraud, also known as 419 scams or Nigerian prince scams, is a type of fraudulent scheme where scammers deceive individuals into paying an upfront fee or providing personal information with the promise of a much larger financial gain in return. Advance Fee Fraud is a form of confidence trick in which scammers convince victims to send them money or disclose sensitive information under the pretense of securing a substantial reward. The "advance fee" refers to the initial payment or fee requested by the fraudsters before the promised windfall can be obtained. These scams often play on the victim's willingness to believe in lucrative opportunities or their sympathy for someone in distress.
 - Nigerian Prince Scam: One of the most well-known examples of Advance Fee Fraud involves an email or letter from a person claiming to be a Nigerian prince, government official, or wealthy individual with large sums of money trapped in a foreign bank account. The scammer requests the victim's assistance in transferring the funds out of the country. In return, the victim is promised a significant percentage of the sum as a reward. However, to proceed with the transaction, the victim is asked to provide personal information or pay various fees, such as legal fees, transfer

charges, or bribes. Example: "Dear Sir/Madam, I am Prince John Doe from Nigeria. I have \$10 million USD that I need to transfer to a foreign account urgently. I need your help, and in return, I will share 30% of the funds with you. Please provide your bank details and send \$5,000 for legal fees to initiate the process."

- Lottery or Inheritance Scam: Scammers may send emails or letters informing recipients that they have won a large sum of money in a lottery or are entitled to a significant inheritance. The victims are instructed to pay upfront fees or taxes to claim the winnings or access the inheritance. These fees are often justified as covering legal expenses, processing fees, or taxes. Example: "Congratulations! You have won the international lottery worth \$1 million. To claim your prize, please send \$5,000 for administrative charges and taxes. Once we receive the payment, we will release the funds to you."
- Job Opportunity Scam: Fraudsters may pose as employers offering lucrative job opportunities, often involving remote work or high-paying positions. The victims are required to pay for work permits, visa processing, background checks, or training materials before they can start the job. However, there is no actual job, and the scammers disappear once the fees are paid. Example: "We have reviewed your application and are pleased to offer you a work-from-home opportunity with a monthly salary of \$5,000. To secure the position, please send \$500 for processing your work permit and training materials."
- Romance Scam: Scammers create fake profiles on dating websites or social media platforms, establish romantic relationships with their targets, and then ask for money under various pretexts such as travel expenses, medical emergencies, or financial hardships. They often claim to be in a difficult situation and seek the victim's sympathy and assistance. Example: "Darling, I am deeply in love with you and want to be together forever. Unfortunately, I have fallen on hard times and need \$2,000 urgently to pay for my mother's medical treatment. Please send the money as soon as possible, and we can start our life together."
- **Malware-Infected Emails:** Malware-infected emails, also known as malicious emails, are messages that contain attachments or links that, when interacted with, can download and install malicious software (malware) on the recipient's device. Malware-infected emails are specifically designed to exploit human vulnerabilities, relying on social engineering techniques to trick recipients into taking actions that compromise their device's security. These emails often employ compelling subject lines, urgent messages, or mimicry of trusted sources to entice users to interact with the malicious content. The attachments may contain executable files, such as .exe or .zip files, or the links may direct users to infected websites where malware is hosted.
 - Fake Invoice or Delivery Notification: Scammers may send emails claiming to be from well-known shipping companies or online marketplaces, providing details about an invoice or delivery status. The email may contain an attachment or link that, when clicked, downloads malware

onto the recipient's device. Example: "Your Recent Purchase Invoice" or "Delivery Status Update: Click Here to Track Your Package"

- Phony Financial Statements: Emails disguised as financial statements or tax documents may be sent, appearing to originate from banks, financial institutions, or accounting services. The attachments or links in these emails can unleash malware, allowing cybercriminals to gain unauthorized access to the victim's financial information. Example: "Important Tax Document Enclosed" or "Year-End Financial Statement: Open Attachment for Details"
 - Job Application or Resume Scams: Malicious emails pretending to be job applications or resumes may contain attachments or links that, when accessed, download malware onto the victim's device. These emails exploit the recipient's curiosity or interest in employment opportunities to trick them into opening the infected attachment or link. Example: "Job Application - Your Dream Opportunity Awaits!" or "Impressive Resume Attached - Open to Learn More"
 - Phishing Emails with Malware Payloads: Some phishing emails include malware-infected attachments or links that are disguised as legitimate messages from trusted organizations. These emails often employ social engineering techniques to persuade recipients to interact with the malicious content, leading to malware installation. Example: "Urgent Account Security Update Required - Click Here to Verify" or "Suspicious Activity Detected - Open Attachment for Details"
 - Fake Software Updates: Cybercriminals may send emails posing as software companies, notifying recipients of critical updates or patches for popular applications. The email may contain a link or attachment that, when accessed, downloads malware instead of the promised software update. Example: "Software Update Required - Click Here to Install the Latest Version" or "Security Patch Released - Download Now to Stay Protected"
- **Fake Lottery or Prize Emails:** Fake lottery or prize emails are deceptive messages that claim the recipient has won a significant amount of money, a lottery jackpot, or an extravagant prize. Fake lottery or prize emails are carefully crafted to appear as legitimate notifications from well-known lotteries, sweepstakes, or organizations. They exploit the recipients' desire for financial gain or the allure of winning valuable prizes. These emails typically inform the recipients of their supposed winnings and urge them to respond promptly to claim the prize. However, to proceed with the prize collection process, scammers often require the victims to provide personal information, pay processing fees, or share bank account details.
 - Lottery Jackpot Notification: The email claims that the recipient has won a substantial amount of money in an international lottery. It may mention the name of a well-known lottery organization and provide a fabricated winning ticket number. The recipient is instructed to respond with personal details and pay administrative or transfer fees to claim the winnings. Example: "Congratulations! You have won \$1,000,000 in the

Global Lotto Jackpot. To receive your winnings, please provide your full name, address, and a processing fee of \$500.”

- Online Sweepstakes Prize: The email informs the recipient that they have been selected as the winner of an online sweepstakes or contest. It often includes the name of a popular brand or a well-known company to appear legitimate. To claim the prize, the victim is asked to share personal information or pay taxes or handling charges. Example: ”You are the lucky winner of our Annual Online Sweepstakes! Claim your prize of a luxury vacation package by providing your contact details and paying a processing fee of \$200.”
 - Inheritance or Donation Windfall: The email claims that the recipient is entitled to a substantial inheritance or a large sum of money from a deceased person or a generous donor. Scammers often create fictional stories about wealthy individuals or beneficiaries who wish to distribute their wealth. The victim is asked to share personal information, pay legal fees, or provide bank account details for the funds to be transferred. Example: ”You have been named the beneficiary of a generous donation! To receive \$5,000,000, please send your full name, address, and a legal fee of \$1,000 to process the transaction.”
 - Prize from a Random Draw: The email states that the recipient’s email address has been randomly selected as the winner of a valuable prize or a gift voucher. The email may claim to be from a popular retailer or an online marketplace. To claim the prize, the victim is requested to provide personal information or pay a small fee for shipping or handling. Example: ”Congratulations! You have won a brand-new smartphone in our monthly lucky draw. To claim your prize, click the link below and pay a small shipping fee of \$10.”
- **Chain Letters:** Chain letters are messages or emails that encourage recipients to pass on the message to multiple other individuals in a sequential manner. Chain letters are a form of correspondence that circulates among individuals, urging them to forward the message to a specific number of people or risk negative consequences or miss out on potential rewards. The letters often claim that by following the chain and forwarding the message, the recipient will receive good luck, financial gain, or blessings. They can be sent through traditional mail, email, or social media platforms, taking advantage of the ease of forwarding and sharing content.
 - Good Luck Chain Letters: These chain letters claim that by forwarding the message to a certain number of people, the recipient will receive good luck or positive outcomes in their life. The letters often state that breaking the chain will result in bad luck or missed opportunities. Example: ”Forward this email to ten people within the next hour, and you will receive unexpected good luck. If you break the chain, you will have bad luck for the next ten years.”
 - Blessing Chain Letters: These chain letters play on the recipients’ religious or spiritual beliefs by promising blessings or divine favor if the message is shared with others. They often state that failing to forward

the message will result in missed blessings or divine disapproval. Example: "Share this message with five friends, and within seven days, you will receive a special blessing from above. If you ignore this message, you will miss out on divine favor."

- Financial Chain Letters: Financial chain letters exploit people's desire for wealth or financial gain. They claim that by participating and forwarding the letter, the recipient will receive a large sum of money or an opportunity for financial success. Breaking the chain is often said to result in missed financial opportunities or financial misfortune. Example: "This chain letter has been circulating for years, and those who participate have received millions of dollars. Send \$10 to the first five names on the list, add your name to the bottom, and forward this letter to ten others. You will soon receive unexpected wealth. If you break the chain, financial misfortune may follow."
- Threatening Chain Letters: Some chain letters use fear tactics to compel recipients to forward the message. They claim that breaking the chain will result in negative consequences, such as accidents, illness, or misfortune. These letters exploit recipients' fear and anxiety. Example: "If you do not forward this message to at least fifteen people, something terrible will happen to you or your loved ones within 24 hours. Take this seriously and don't risk the consequences."
- **Stock Pump-and-Dump:** Stock Pump-and-Dump is an illegal investment scheme in which scammers fabricate the price inflation of a stock, by laying out incorrect and deceiving information, and then scam unsuspecting investors by selling stock shares to them at the inflated price. Stock Pump-and-Dump is a manipulation technique used by individuals or groups to artificially increase the price of a stock for personal gain. The scheme typically involves the following steps:
 - Promotion: The fraudsters behind the scheme promote a particular stock by disseminating false or misleading information about the company. This can be done through various channels, including social media, online forums, email newsletters, or fake press releases. The information may include positive news, exaggerated claims about the company's prospects, or endorsements from fictitious experts.
 - Buying: As the false information begins to circulate, unsuspecting investors are enticed to buy the stock, believing that they are getting in on a promising investment opportunity. The increasing demand for the stock drives up its price.
 - Selling: Once the stock price has been artificially inflated, the fraudsters sell their own shares at the elevated price, realizing substantial profits. These sales often occur in large volumes, flooding the market and creating an illusion of high trading activity.
 - Dumping: After the fraudsters have sold off their shares, the demand for the stock drops significantly. As a result, the price plummets, causing losses for those who bought the stock at the inflated price.

- Penny Stock Promotion: Fraudsters may target penny stocks, which are low-priced and thinly traded stocks, as they are more susceptible to manipulation. The promoters may use false claims about the company’s breakthrough technology, upcoming partnerships, or anticipated regulatory approvals to create hype and attract investors. Once the stock price surges, they sell their shares and abandon the stock, causing it to collapse.
 - Online Chatroom Scheme: Fraudsters may join online chatrooms or discussion forums dedicated to stock trading and recommend certain stocks as ”hot tips” or ”hidden gems.” They may fabricate positive news, post false financial reports, or create a sense of urgency to entice others to buy the recommended stocks. After the price has risen significantly, they sell their shares, leaving other investors with worthless stocks.
 - Pumping through Social Media: Fraudsters may use social media platforms, such as Twitter, Facebook, or Reddit, to disseminate false information and generate buzz around a particular stock. They may create multiple accounts or employ bots to amplify the messages, making them appear more credible and widespread. Unsuspecting investors who follow the recommendations and buy the stock end up losing money when the price crashes.
 - Insider Pump-and-Dump: In some cases, insiders with privileged information about a company may participate in a pump-and-dump scheme. They use their knowledge to promote the stock and attract investors. Once the price reaches a peak, they sell their shares, taking advantage of the artificial price increase and leaving other investors at a disadvantage.
- **Dating or Romance Scams:** Dating or romance scams are fraudulent schemes where individuals, often posing as potential romantic partners, exploit the emotions and trust of unsuspecting victims for financial gain. Dating or romance scams involve scammers who create fake identities and profiles on dating websites, social media platforms, or online forums. They use these profiles to establish romantic relationships with their victims, often targeting individuals who are seeking companionship, love, or emotional support. The scammers invest time and effort into building trust and emotional connections with their targets, only to exploit them for financial gain. They may employ various tactics, such as professing love quickly, making future plans, or fabricating stories of financial hardship or emergencies, to manipulate their victims into sending money or providing personal information.
 - Catfishing Scams: Scammers create fake profiles using stolen or stock photos, often portraying themselves as attractive individuals. They initiate contact with potential victims and develop online relationships, gradually building trust and emotional intimacy. Once a connection is established, they may ask for financial assistance, claiming to be in a difficult situation or facing an emergency. Example: A scammer creates a fake profile on a dating site, using a photo of an attractive person. They engage in conversation with their target, sharing personal stories and expressing affection. After a few weeks, they claim to be in a financial crisis and ask the victim to send money to help them overcome the situation.

- **Military Romance Scams:** Scammers impersonate military personnel, using stolen photos and fabricated stories about serving in the armed forces. They exploit the respect and admiration society has for military personnel to gain sympathy and trust from their victims. These scammers often claim to be deployed overseas and request money for supposed emergencies or travel expenses to meet the victim. Example: A scammer poses as a soldier deployed in a foreign country. They initiate contact with their target and establish an emotional connection by sharing stories of their military service and expressing feelings of love. They eventually ask the victim for financial assistance, stating that they need funds for medical treatment or to arrange for leave from their duties.
- **Online Dating Extortion:** Scammers establish relationships with their victims and engage in explicit or intimate conversations, often through video calls or sharing compromising photos. They then use these private materials as leverage to extort money from the victims, threatening to share the content with their family, friends, or colleagues. Example: A scammer gains the trust of their target through an online dating platform. They engage in explicit conversations and convince the victim to share compromising photos or videos. Subsequently, the scammer threatens to release the materials publicly unless the victim pays a significant sum of money.
- **Advance Fee Fraud in Relationships:** Scammers exploit the emotional vulnerability of their victims and gradually introduce financial requests under the guise of helping the relationship progress. They may ask for money to cover travel expenses, visa fees, or customs charges, promising to meet the victim in person or relocate to their country. Example: A scammer builds a relationship with their victim, expressing a desire to meet in person. They claim to require financial assistance to obtain travel documents, pay for flights, or clear customs. They assure the victim that the money will be repaid upon arrival but disappear after receiving the funds.
- **Medication Spam:** Medication spam refers to unsolicited emails or messages that promote or sell prescription drugs or other medications. Medication spam involves the distribution of unsolicited emails, messages, or advertisements that promote the sale of medications, often including prescription drugs. These spam messages are typically sent en masse to a wide range of recipients, regardless of whether they have expressed interest or consented to receive such communications. The primary goal of medication spam is to generate sales for unscrupulous online pharmacies or sellers who may operate illegally, selling counterfeit or substandard medications. Medication spam can be a significant public health concern as it promotes the misuse of drugs and poses risks to consumers who may unknowingly purchase unsafe or ineffective products.
 - **Prescription Drug Offers:** Medication spam often includes offers for prescription drugs that require a valid prescription from a healthcare professional. These messages may claim to provide access to medications without requiring a prescription or may offer to arrange online consultations

with doctors who will issue prescriptions without a proper evaluation. Example: "Buy Viagra without a prescription! Get the best deals on Viagra, Cialis, and Levitra. No prescription is needed. Order now!"

- Online Pharmacy Promotions: Medication spam frequently promotes online pharmacies that claim to offer a wide range of medications at discounted prices. These messages may promise convenience, privacy, and quick delivery. However, the legitimacy and safety of such pharmacies may be questionable, as they may operate without proper licensing or quality control measures. Example: "Buy your medications online! Get up to 70% off on brand-name and generic drugs. Fast shipping and discreet packaging. Visit our online pharmacy today!"
 - "Miracle" or Unproven Remedies: Some medication spam messages exploit individuals seeking unconventional or alternative remedies by promoting unproven "miracle" drugs or treatments for various health conditions. These messages often make extravagant claims about the effectiveness of the products, targeting vulnerable individuals who may be desperate for a solution. Example: "Discover the secret to curing cancer! Our revolutionary herbal remedy guarantees complete remission. Order now and reclaim your health!"
 - Weight Loss Supplements: Medication spam frequently advertises weight loss supplements or diet pills that promise rapid and effortless weight reduction. These messages exploit individuals' desires for quick fixes and may use before-and-after photos or testimonials to create a sense of credibility. Example: "Lose 20 pounds in two weeks! Try our breakthrough weight loss pill and achieve your dream body. Limited time offer. Order now!"
- **Political or Ideological Spam:** Political or ideological spam refers to unsolicited messages or communications that aim to promote or advocate for specific political or ideological beliefs, often with the intention of influencing public opinion or advancing a particular agenda. Political or ideological spam involves the unsolicited distribution of messages, content, or propaganda that promotes specific political or ideological beliefs. The primary objective is to influence public opinion, recruit supporters, or rally individuals around a particular cause or agenda. Political or ideological spam can take various forms, ranging from biased news articles and manipulated images to viral social media posts and email campaigns. The messages often target specific demographics or individuals who are likely to align with the promoted ideology or political agenda.
 - Misinformation Campaigns: Political or ideological spam may involve the deliberate spread of false or misleading information to advance a particular narrative or discredit opposing viewpoints. This can include the dissemination of fabricated news articles, manipulated images, or videos that aim to deceive or manipulate public perception. Example: A spam campaign targeting voters during an election may distribute false information about a candidate's criminal record, financial impropriety,

or involvement in scandalous activities to damage their reputation and sway public opinion against them.

- **Partisan Social Media Posts:** Political or ideological spam often manifests as partisan or biased content shared on social media platforms. These posts aim to mobilize supporters, reinforce existing beliefs, or provoke emotional responses from individuals who align with a particular political ideology. Example: A spam account on a social media platform may consistently share posts and memes that praise one political party or ideology while demonizing opposing viewpoints. These posts may contain cherry-picked facts, inflammatory language, or exaggerated claims to appeal to like-minded individuals and generate engagement.
- **Advocacy Emails:** Political or ideological spam can also be in the form of unsolicited emails that promote specific political causes, ideologies, or advocacy campaigns. These emails may seek donations, call for action, or encourage recipients to support certain policies or candidates. Example: An email campaign may be launched to solicit donations for a political organization advocating for a particular social or environmental issue. The emails may use emotionally charged language, highlight urgent threats, or appeal to recipients' values to persuade them to contribute financially.
- **Astroturfing:** Astroturfing refers to the creation of fake grassroots movements or online communities that appear to be organic but are actually orchestrated by political or ideological actors. These campaigns aim to simulate widespread public support for a particular cause or agenda. Example: A political group may create numerous social media accounts posing as ordinary citizens to flood comment sections, discussion forums, or online polls with messages supporting a specific policy or candidate. The goal is to create an illusion of broad support and influence public perception

2.5 Previous Work Related to Email Spam Detection and categorization

2.5.1 Related work in email categorization [3]

- **Prior Email Prioritization**

Horvitz et al. developed an email alerting system that uses Support Vector Machines to categorize freshly incoming email messages into two categories, namely high or low in terms of utility. Along with the system's predictions, probabilistic ratings were also offered. Personalization, on the other hand, was not taken into account in their technique, and priority modeling and social network analysis were not among their technical interests. Hasegawa and Ohara advocated using Linear Regression with two levels of evaluation. They extracted characteristics using over a thousand rules. Despite the fact that they stated that the priority should be individualized, they only tested their concept on one user. There was no comprehensive examination of alternative priority modeling methodologies or social network analysis.

- **Social Clustering**

Tyler et al. used the Newman clustering technique to automatically detect social patterns in email conversations. They discovered that the automatically identified social structures are very close, if not identical, to human interpretations of organizational systems. They also identified social leaders using email social networks. However, they did not prioritize email messages using social network analysis (clusters or leadership ratings).

Gomes et al. utilized email messages to automatically classify people into two groups: sender clusters and receiver clusters. The senders were grouped according to the similarity of their receiver lists, and the receivers were clustered according to the closeness of their sender lists as well; email messages were not utilized.

Author-RecipientTopic (ART) model developed by McCallum et al. models the linkages between sender and receivers as well as direction sensitive topic distribution based on Latent Dirichlet Allocation (LDA). We were able to determine the probabilistic topic distribution based on the relationships between persons using the ART model. The ART model was then expanded to incorporate social roles, giving rise to the Role-ART (RART) paradigm.

Johansen et al. developed a social clustering technique to email message significance prediction. They collected email data from many users and created user social groupings. Some clusters are considered "essential" for each user, while others are not. The relevance of each test instance of an email message is anticipated based on its sender's cluster membership: If the sender is a member of an important cluster, the message is considered significant; otherwise, it is expected to be unimportant.

- **Social Importance Metrics**

In email research, many social measures have been employed. Neustaedter et al. created metrics for quantifying persons' social relevance based on observations in email fields such as from, to, and cc, as well as recorded activities such as replying and reading. Instead of prioritizing incoming email messages, they utilized these metrics to retrieve old email messages. Martin et al. utilized each person's out-degree (the number of distinct receivers) and in-degree (the number of distinctive senders) in an email social network to discover worms that spread via email messages.

2.5.2 Existing work related to email spam detection

the Table 2.1 shows work previously done by [9] related to spam detection

Author and Year	Dataset Used	Method Used	Evaluation Parameter	Remark
Renuka and Visalakshi (2014)	Ling Spam Email Corpus	Support Vector Machine (SVM) with Latent Semantic Indexing (LSI)	Precision, recall and accuracy	Proposed SVM-LSI performs well in comparison with other state of art research
Harisingh aney et al. (2014)	Enron Corpus dataset	Naive Bayes, KNN algorithm and Reverse DBSCAN algorithm	Precision, sensitivity, accuracy and specificity	Results with pre-processing steps were reported to be better than results without pre-processing steps. However, authors record significant time consumption for data filtration.
Idris et al. (2015)	Spam base dataset	Particle Swarm Optimization and Negative Selection Algorithm	Accuracy, F-measure, Negative prediction value and Statistical t-test, sensitivity, correlation factor, specificity and positive prediction value	Overall, the NSA-PSO achieved better accuracy results than the NSA.
Mohamad and Selamat (2015)	Manually Generated Dataset	Rough set theory and Term Frequency Inverse Document Frequency	Classification accuracy	Instead of email spam classification, the main emphasis was on feature extraction
Tuteja and Bogiri (2016)	Manually Generated Dataset	Artificial Neural Network and K-means Clustering	Recall and Precision	Better results were observed with preprocessing steps when compared to results without preprocessing
Kaur and Sharma (2016)	Spam base dataset	Decision Tree Algorithm and Integrated Concept of PSO	Mean absolute error, F-measure and correctly classified ratio	No information available regarding the use of the feature extraction.

Feng et al. (2016)	DATA MALL (Chinese Spam Email Dataset)	Integrated SVMNB (Support Vector Machine -Naïve Bayes)	Recall, precision and execution time	When compared to individual SVM and NB approaches, the integrated approach yields better results.
Kumaresan and Palanisamy (2017)	Ling Spam dataset	Stepsize Cuckoo Search with Support Vector Machine	Specificity, sensitivity and accuracy	Overall, the modified algorithm outperforms the original CS in terms of classification speed.
Olatunji et al. (2017)	Text Corpus Spambase dataset	Support Vector Machines and Extreme Learning Machines	Time taken and accuracy	SVM outperforms ELM in terms of accuracy, but ELM takes less time than SVM.

Table 2.1: Previous Work Related to Email Spam Detection

Chapter 3

Initial work plan and proposed methodology

To begin with, at first, we considered the E-mail from the dataset in raw format. In order to convert the existing raw dataset to a usable dataset for our algorithms, we first have to apply some preprocessing methods [14].

- **Tokenization:** Tokenization works by removing data from the environment and replacing it with tokens in this case breaking the stream of text in the E-mail into tokens of individual words.
- Removal of stop words such as “a”, “an”, “the” etc.
- **Lemmatization:** Lemmatization usually refers to identifying the inflected forms of a word and returning its base form (e.g. “better” is lemmatized as “good”). In this case group together derivationally related words with similar meanings by morphological analysis.
- **Stemming:** Stemming refers to removing or replacing word suffixes (e.g. “running” is stemmed as “run”) and identifying the common root form of a word. In this case, bringing the tokens obtained from the previous step to their root form.

We then take the preprocessed dataset and apply Correlation Based Feature Selection (CFS) approach, in order to reduce dimensionality and select only relevant feature words from the data we need to apply a correlation-based feature selection approach.

$$CFS = \max_{sk} \left[\frac{r_{cf1} + r_{cf2} + \dots + r_{cfk}}{\sqrt{k + 2(r_{f1f2} + \dots + r_{fifj} + \dots + r_{fkfk-1})}} \right]$$

Now we split our dataset into a training dataset (66%) and a test dataset (34%) and start training the training dataset with the help of Naive Bayes and Particle Swarm Optimization (PSO).

At first, using Naive Bayes we will find the probability distribution of the tokens with the selected feature. The formula used for calculating the probability distribution is,

$$p(y|(f_1, f_2, f_3, \dots, f_n)) = \frac{p((f_1, f_2, f_3, \dots, f_n)|y)p(y)}{p(f_1, f_2, f_3, \dots, f_n)}$$

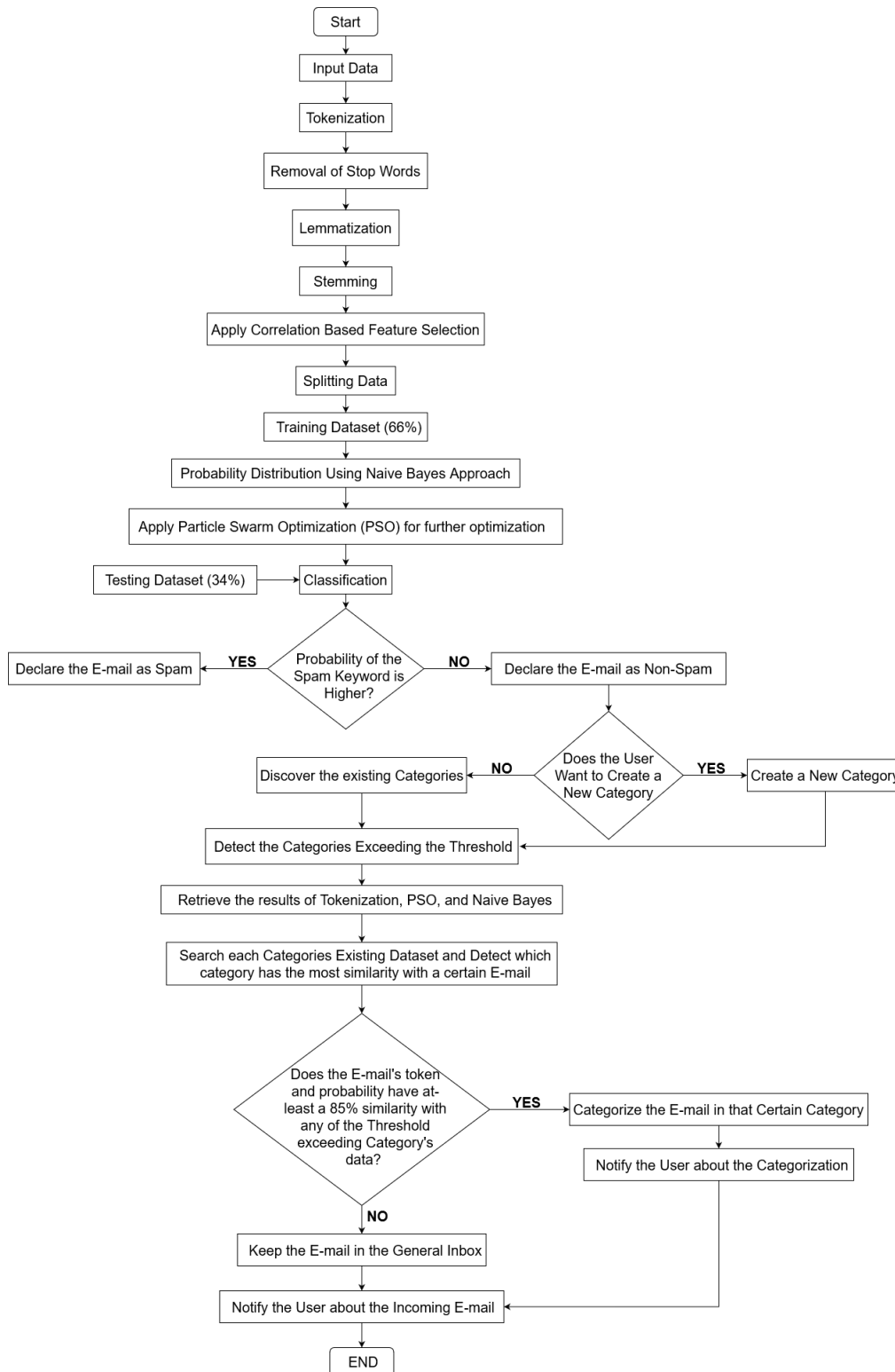


Figure 3.1: Initial Work flow of our system

Here, f is the feature vector set $(f_1, f_2, f_3, \dots, f_n)$, the class variable form- possible outcomes are defined as y , $P(y|x)$ is the posterior probability and $P(yx)$ is dependent for any particular class of $P(x|y)$, $P(x)$ is evidence depending on the feature, $P(y)$ is prior probability.

After determining the probability distribution of the tokens now we can apply PSO to optimize our outcome. We consider every tokens as particles and initially these particles will randomly search for food source as the best feature match for tokens and then it will search for Local and Global solutions. The performance of each particle will dependent on the similarity from which feature has to be optimized. Each of the particles searches over n - dimensional search space and will update the following information:

- X_i - current position of particle x
- P_i - personal best position of particle x
- V_i - current velocity of the particle x

The velocity updates of PSO will be calculated using the equation,

$$V_{i(t+1)} = \omega V_{it} + c_1 r_1 (P_{it} - X_{it}) + c_2 r_2 (P_g - x_{it})$$

Here V_i is the new velocity and the position of the particle updates with velocity as,

$$X_{i(t+1)} = X_{it} + V_{i(t+1)}$$

From the following information obtained now we update the position for each particle and store the global best solution. Finally based on evaluated feature similarity (using PSO) each token will be declared as spam or non-spam. Using the classification of the tokens now we can categorize sentences, If the probability of spam tokens is more then we will categorize the sentence as spam otherwise it will be considered as non-spam. Thus from the categorization of sentences now, we can categorize a complete E-mail in the same procedure. After successfully training the dataset with the previously mentioned algorithms now we can start working with our test dataset. After the categorization of whether the Email is spam or not, we can categorize the non-spam data into various categories depending on the users' desire (users can create desired categories based on their needs). For every Category there will be a specific threshold, for example, our model will have categories in only the categories that have 20 or more (threshold) emails in them. So we will first discover the categories and detect which categories are exceeding the threshold. We then retrieve the results of Naive Bayes and PSO for each token and check with every eligible category how much percentage it has similarity with that certain E-mail. We will then choose the category that has the most similarity percentage and check if it has an 85% similarity or not. If it has a similarity percentage lower than 85% then that email will not be categorized in any specific category rather it will just remain in the general inbox.

3.1 Preliminary Analysis

According to the research work [5], after experiments are carried out with four classifier algorithms, the Random Forest algorithm is observed to have a higher

accuracy among all, even Naive Bayes. However, we have chosen to use the Naive Bayes algorithm in our model to classify the E-mails because the Random Forest algorithm shows inefficiency in predicting during the test phase, as mentioned in this article [16]. As a higher accuracy of the algorithm comes with the cost of slow performance, we opt for the Naive Bayes algorithm instead.

The research work of [6] compares various machine learning algorithms to see which one is the most optimized and provides a comparatively better solution to our problem. The classifiers that were compared here were Bayesian network, Naive Bayes, Support Vector Machine (SVM), Decision Tree (J48), and Boosting with AdaBoost. These classifiers were compared using 3 instruments,

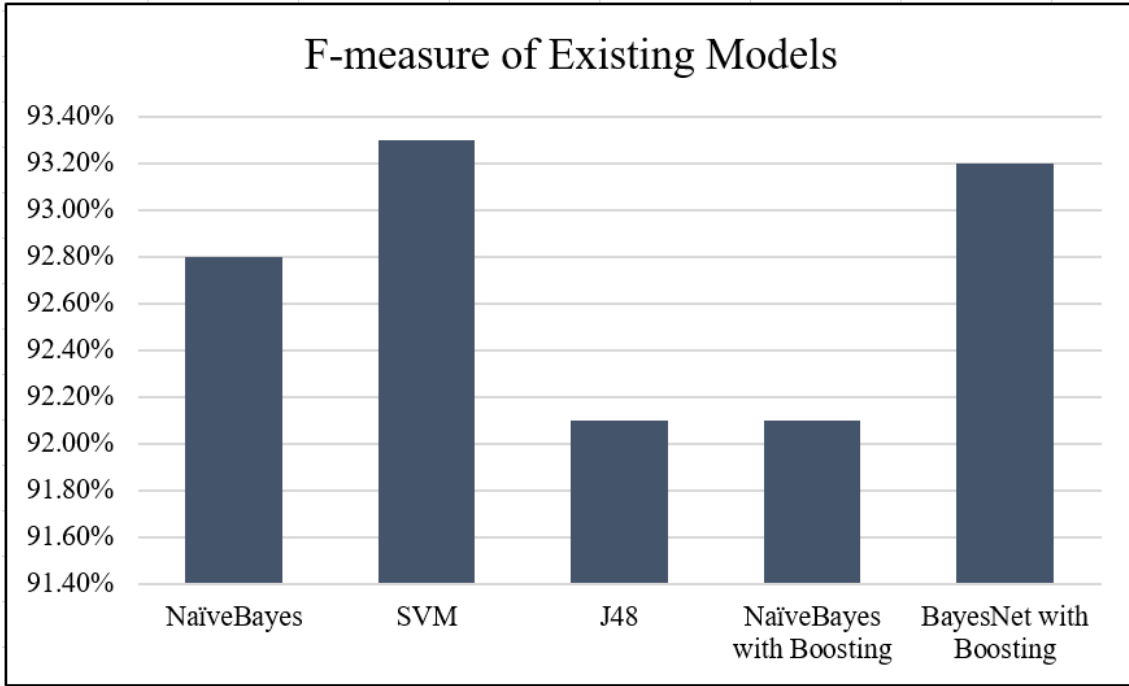


Figure 3.2: F-measure of the existing models

1. **F-measure:** This measures the accuracy of a test using two parameters precision (P) and recall (R) to compute the score, according to the work [6]. The mathematical equation is,

$$F_{h,s}^{Value} = \frac{2 \times Precision_{H,S} \times Recall_{H,S}}{Precision_{H,S} + Recall_{H,S}}$$

Here, Precision (P) is the number of accurate positive outputs divided by the number of all positive outputs, and recall (R) is the number of accurate positive outputs divided by the number of positive outputs that should have been returned. The results can be seen in figure 3.2

2. **False Positive Rate:** FP rate probability of falsely categorizing an E-mail as spam (in other words Ham). It can be calculated using the mathematical equation,

$$FP_{rate} = \frac{h_{am}^{mis}}{h_{am}^{mis} + h_{am}^{correct}}$$

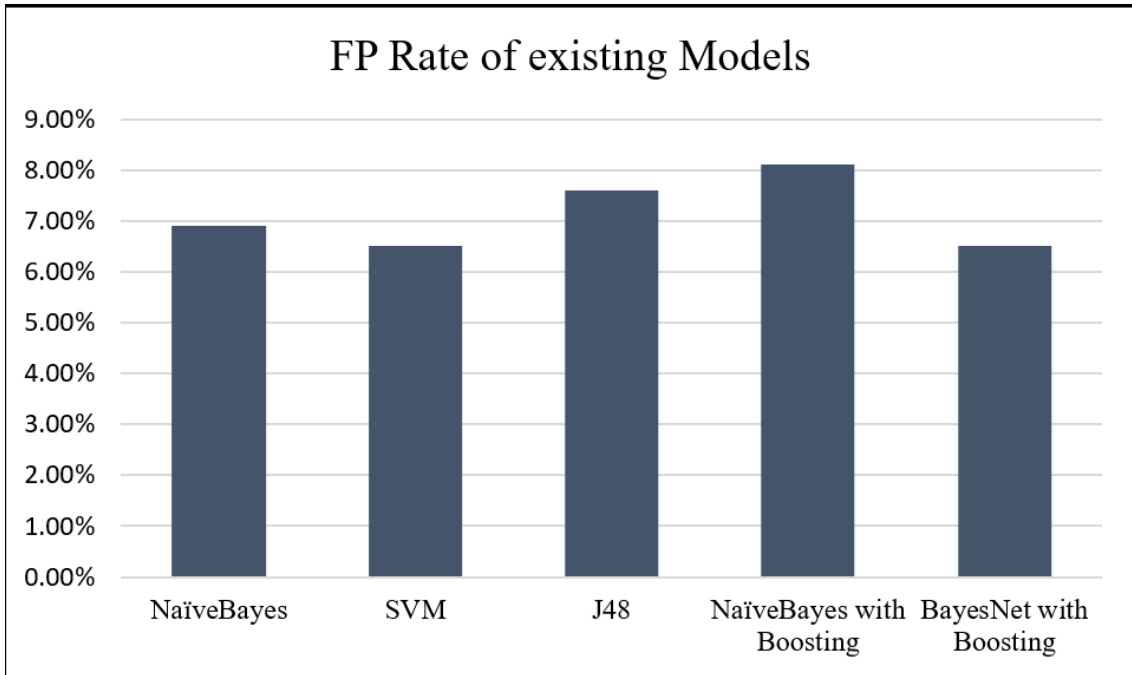


Figure 3.3: FP rate of the existing models

	FP rate	F-measure	Training time
Bayes	8.1%	92.0%	0.51 sec
NaiveBayes	6.9%	92.8%	0.25 sec
SVM	6.5%	93.3%	10.54 sec
J48	7.6%	92.1%	4.73 sec
With Boosting			
BayesNet	8.1%	92.1%	7.18 sec
NaiveBayes	6.5%	93.2%	17.35 sec

Table 3.1: comparison between different classifiers on a test dataset. (from [6])

3. **Training Time:** the time required to Train/Test a dataset.

Here, table 3.1 shows that SVM and Ada boosting on Naive Bayes gives us the least false positive rate, SVM gives the best false positive rate and Naive Bayes gives the least training time. Thus, no classifier alone can provide an F-measure close to 100%.

Moreover, we chose to use the logistic regression model as our initial approach because logistic regression models are best suited for binary classification models. As our system is concerned with the prediction of emails as either spam or non-spam mail, these two categories can be represented as a binary classification system where 0 represents spam mail and 1 represents non-spam mail.

Although a logistic regression model seemed best suited for our system, due to a failure of a correct prediction, we cannot use this regression model for classification.

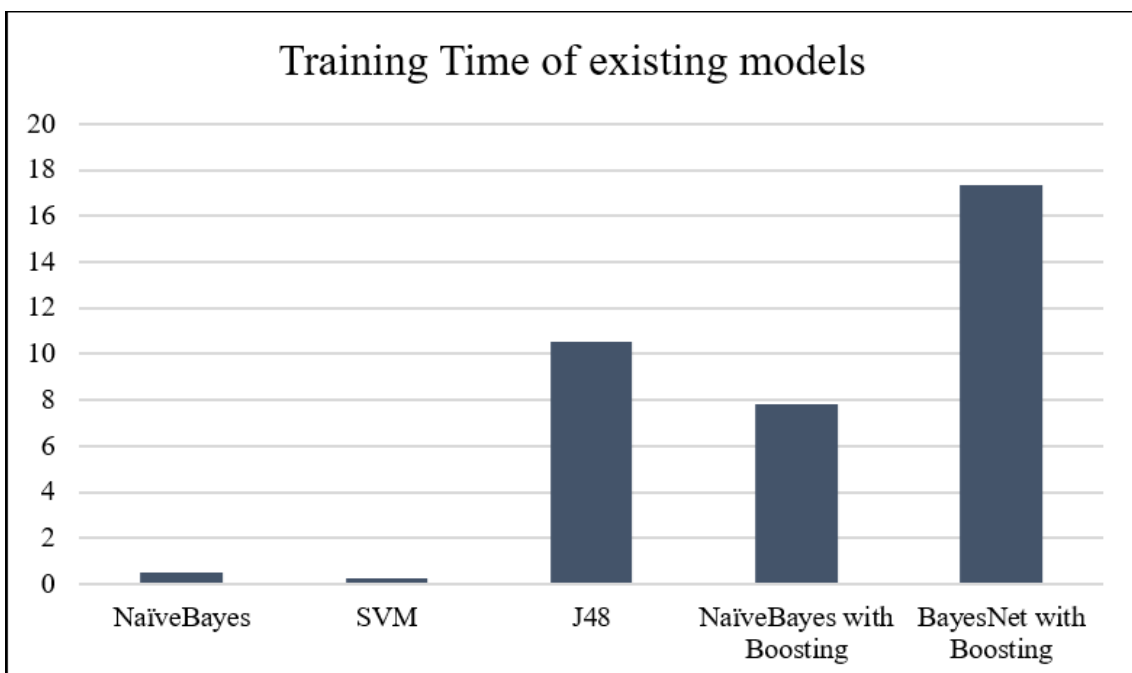


Figure 3.4: Training time of the existing models

Chapter 4

Working with Dataset

4.1 Dataset Description and Collection

We commenced the process of gathering our primary dataset through the distribution of a Google form, where individuals were requested to furnish us with their spam and non-spam emails. These submissions could be made via their mobile devices or personal computers. We have collected 1741 emails using this method of collection. In our primary dataset, there are 860 spam emails and 880 nonspam emails. We have also taken data from a secondary dataset of enron corp. from Kaggle which contained a total of 5574 emails. Among them, 1750 are spam emails and 3824 are nonspam emails. These secondary data were used to check what effect does adding more data or a different variety of data has on the accuracy of our model.

4.2 Data Labeling and Naming Format

We compiled all the spam emails into a spreadsheet with two columns. One was titled subject and contained all of the spam email subjects, while the other was titled body and contained all of the collected spam email bodies. In order to work with the dataset, we converted the spreadsheet to a.csv file and incorporated it into our working model.

4.3 Data Pre-Processing

4.3.1 Convert to LowerCase and Punctuation

Our objective here is to convert our textual data into lowercase and eliminate all the punctuation marks. This step is necessary because when we have a text input, such as a paragraph we have words both in lower and upper case. However, the same words written in different cases are considered different entities by the computer, thereby causing complications. To address this issue, we have converted all the words to lowercase. This ensures uniformity throughout the text. Additionally, the removal of punctuation marks will help in treating each text equally.

4.3.2 Removal of Stop Words

Stop words are a compilation of words that come frequently in any language. However, they do not add much meaning to the sentences. These are common words that are part of the grammar of any linguistic system. Every language possesses its own distinct list of stop words. Due to the limited significance of stop words in the overall meaning of the sentence, we excluded these words from our textual data. This helps in dimensionality reduction by eliminating unnecessary information. This is extremely useful for large datasets.

4.3.3 Stemming

Stemming refers to the procedure of reducing a word to its fundamental root or stem. This includes the removal of affixes from the word, thereby isolating it and keeping it in its root form or lemma.

4.3.4 Tokenization

Tokenization is the process of the systematic fragmentation of the original text into discrete units known as tokens for further analysis. Tokens represent different pieces of the original text; however, keep in mind they are not broken down into a base form. This process is significant because the meaning of the text can be interpreted through analysis of the words present in the text.

4.3.5 Lemmatization

The process of stemming does not guarantee words that are part of the language vocabulary. It often results in words that have no meaning to the users. To address this issue, we use the concept of lemmatization. Unlike stemming, lemmatization allows for more accurate word transformation by considering the context in which the words appear. In the case of lemmatization, we can pass a POS parameter. This is used to provide the context in which we wish to lemmatize our words by mentioning the Parts Of Speech (POS). If no POS is mentioned, the default assumption is that the words are nouns.

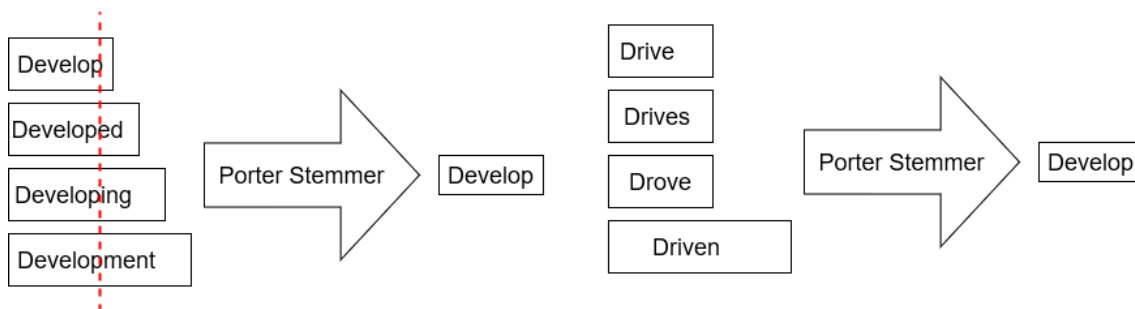


Figure 4.1: Stemming and Lemmatization

Chapter 5

Working Model and Implementation

5.1 Spam Detection Using Naive Bayes Theorem

In our current working model for spam detection, we start coding by importing libraries. We import the CSV, numpy, pandas, train test split, TfidfVectorizer, Logistic Regression, accuracy score, string, stopwords, scipy stats, seaborn, chi2.contingency, matplotlib.pyplot, re, nltk, PorterStemmer, WordNetLemmatizer.

CSV is the Comma Separated Values format and it is the most common import and export format for spreadsheets and databases. Numpy stands for Numerical Python which is a scientific computing library built on top of the Python programming language. Pandas is an open-source data analysis library built on Python. We import the train test split from the Sklearn or Scikit-learn Python library. It offers different features for data processing. Model_selection is a method by which we can analyze data and then can use it to measure new data. For accurate results, it is important to select a proper model when making a prediction. In order to accomplish that, we need to train our model by using a specific dataset and afterward, we need to test the model against another dataset. However, in our case, as we have only one dataset we split it by using the Sklearn train_test_split function. The feature extraction module from sklearn can be used to extract features from our dataset in a format that is supported by machine learning algorithms. The TfidfVectorizer converts our dataset into a matrix of TF-IDF features. Logistic Regression is a classification algorithm. It measures the relationship between the categorical dependent variable and one or more than one independent variables by calculating the probability of an event occurring using its own logistic function. One of the significant stages in our model is the accuracy of our model which we predicted using Python's scikit learn library. As we have textual data it was vital for us to import strings. The NLTK corpus is a huge dump of all kinds of natural language datasets and as our data preprocessing step includes the removal of stop words we imported stopwords from the NLTK corpus. As well as we use the Porter stemmer for data mining and information retrieval. Also in order to lemmatize the dataset we use Wordnet lemmatizer. Afterwards, we will be doing CFS for which we imported scipy stats which is a module that contains a large number of correlation functions and chi2.contingency. Furthermore, after implementing CFS we will be creating a

heat map for which we imported the seaborn library and matplotlib which helps in visualization and exploratory analysis.

After that we apply CFS in our dataset. CFS stands for Correlation-based Feature Selection. It is a filtration approach and hence it is independent of the final classification model. The aim of this method is to find the features with high feature class correlation and to eliminate the features with low feature correlation. This helps to eradicate data redundancy in our dataset. We build the Cramer's V function to aid us with correlation. For visual aid lastly created a heatmap.

Afterward, we carried on to data preprocessing where we converted all our data to lowercase and removed all the punctuations, we removed the stop words, we applied stemming, we tokenized all the words and lastly applied lemmatization.

All this helped us to create a vocabulary list. We used this list to create a dictionary and then converted it to the data frame we needed.

Then we split our data into training and testing. For training, we used 80% of the data and for testing, we used 20% of the data.

Now we have to calculate the constants first.

$$P(\text{Spam}|w_1, w_2, \dots, w_n) \propto P(\text{Spam}) \cdot \prod_{i=1}^n P(w_i|\text{Spam})$$

$$P(\text{Ham}|w_1, w_2, \dots, w_n) \propto P(\text{Ham}) \cdot \prod_{i=1}^n P(w_i|\text{Ham})$$

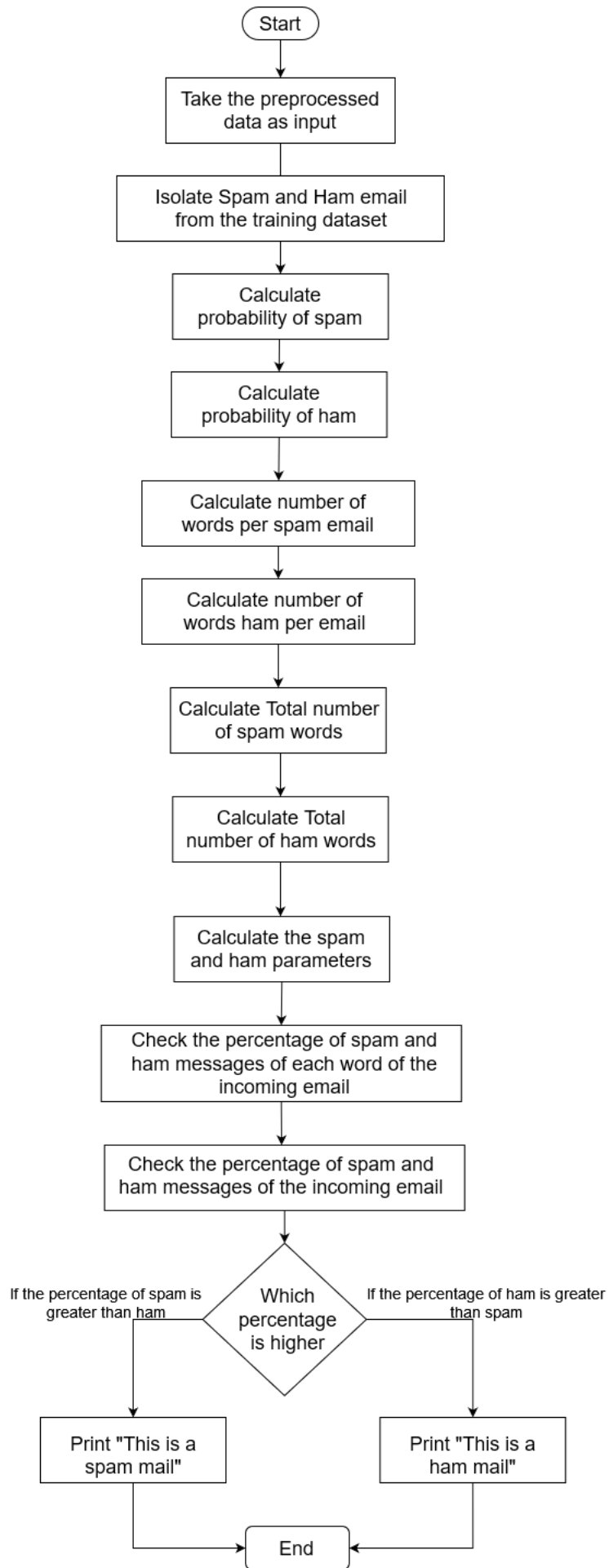
Also, to calculate $P(w_i|\text{Spam})$ and $P(w_i|\text{Ham})$ inside the formulas above, we'll need to use these equations:

$$P(w_i|\text{Spam}) = \frac{N_{w_i} + \alpha}{N_{\text{Spam}} + \alpha \cdot N_{\text{Vocabulary}}}$$

$$P(w_i|\text{Ham}) = \frac{N_{w_i} + \alpha}{N_{\text{Ham}} + \alpha \cdot N_{\text{Vocabulary}}}$$

Some of the terms in the four equations above will have the same value for every new message. we can calculate the value of these terms. We can calculate the value of these terms once and avoid doing the computations again when a new message comes in. To start we will calculate $P(\text{Spam})$ and $P(\text{Ham})$ then calculate $N(\text{Spam})$ and $N(\text{Ham})$. Here $N(\text{Spam})$ is equal to the number of words in all the spam messages, it's not equal to the number of spam messages, and it's not equal to the total number of unique words in spam messages and $N(\text{Ham})$ is equal to the number of words in all the non-spam messages, it's not equal to the number of non-spam messages, and it's not equal to the total number of unique words in non-spam messages. We also use Laplace smoothing and set Equations.

After that, We calculated the parameters using the $P(w_i|\text{Spam})$ and $P(w_i|\text{Ham})$ mentioned above. and classified a new message. To classify a new message, we take



32
Figure 5.1: Work flow of our system

in as input a new message (w_1, w_2, \dots, w_n). Then we calculate $P(\text{Spam}—w_1, w_2, \dots, w_n)$ and $P(\text{Ham}—w_1, w_2, \dots, w_n)$. Next we compare the values of $P(\text{Spam}—w_1, w_2, \dots, w_n)$ and $P(\text{Ham}—w_1, w_2, \dots, w_n)$ and consider following 3 situations:

- If $P(\text{Ham}—w_1, w_2, \dots, w_n) > P(\text{Spam}—w_1, w_2, \dots, w_n)$, then the message is classified as ham.
- If $P(\text{Ham}—w_1, w_2, \dots, w_n) < P(\text{Spam}—w_1, w_2, \dots, w_n)$, then the message is classified as spam.
- If $P(\text{Ham}—w_1, w_2, \dots, w_n) = P(\text{Spam}—w_1, w_2, \dots, w_n)$, then the algorithm may request human help.

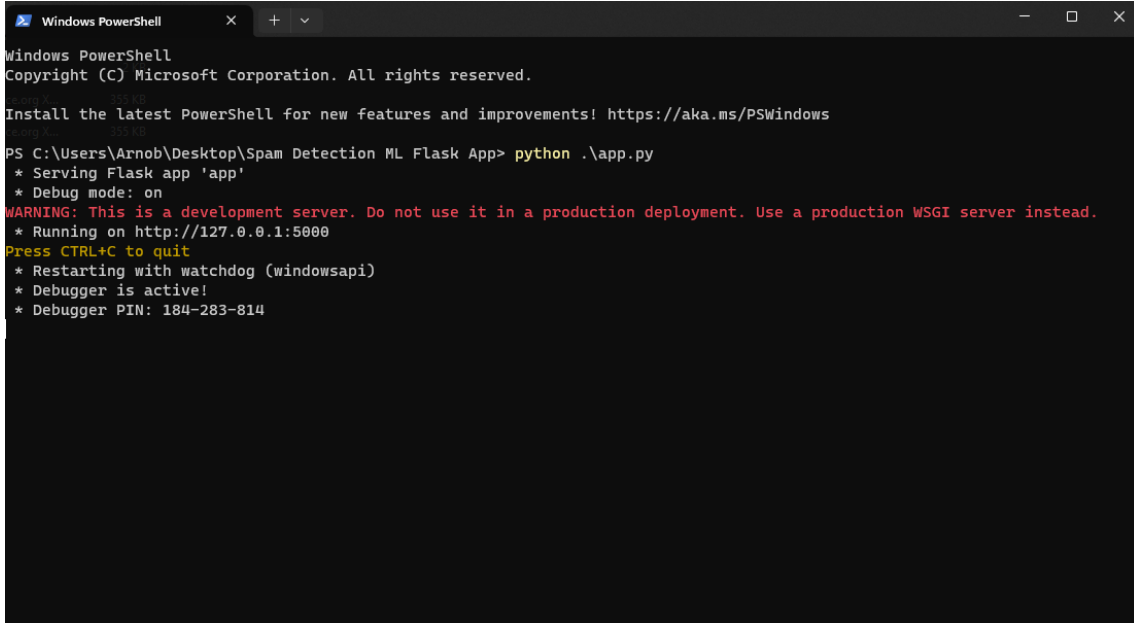
Moreover, if a new message contains words that are not part of the vocabulary, we ignore those words when calculating the probabilities

Using the classification of the tokens now we categorize sentences. If the probability of spam tokens is more than we categorize the sentence as spam otherwise it will be considered as non-spam. Thus from the categorization of sentences now, we can categorize a complete Email in the same procedure. After successfully training the dataset with the previously mentioned algorithms now we start working with our test dataset. After the categorization of whether the Email is spam or not, we categorize the non-spam data into various categories depending on the users' desire (users can create desired categories based on their needs). For every Category there is a specific threshold, for example, our model has categories in only the categories that have 20 or more (threshold) emails in them. So we first discover the categories and detect which categories are exceeding the threshold. We then retrieve the results of Naive Bayes for each token and check with every eligible category how much percentage it has similarity with that certain Email. We then choose the category that has the most similarity percentage and check if it has an 85% similarity or not. If it has a similarity percentage lower than 85% then that email is not categorized in any specific category rather it just remains in the general inbox.

5.2 Web Based Application

The application is built using Python. It uses Flask as a networking framework. Flask is a lightweight WSGI or Web Server Gateway Interface. It tells how a web server and an application will communicate with each other and process requests. Flask is a very simple framework however it is powerful as it can be scaled up to be used in complex applications. The application first runs on the local host computer as a server and opens a website. The application runs from that website as it takes the data and processes it in the background. The back end of the application uses our working model. The front end is made with simple CSS making it very easy to use. The first page is the "home" page where the GUI of the webpage has a text field where the user inputs their email and a simple button to proceed with the filtering process. When the user inputs their email and press predict, then the application will route to "predict" where it will send the email as data to the back-end of the application. There it will pre-process the data and compare it against our model. The result from the comparison will dictate which page comes next by a simple conditional statement. If the mail is found spam then it will lead to "spam" page

which will notify the user their input email is spam and if it is not spam then it will lead to "ham" page which will notify that the email is not spam. Furthermore, when the email is not spam, the user will have the choice to add the email to a category either an existing one or a newly created one. Also when the user gets a spam email they can choose to mark it as not spam or if the email was falsely flagged as one.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Arnob\Desktop\Spam Detection ML Flask App> python .\app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 184-283-814
```

Figure 5.2: Starting the app

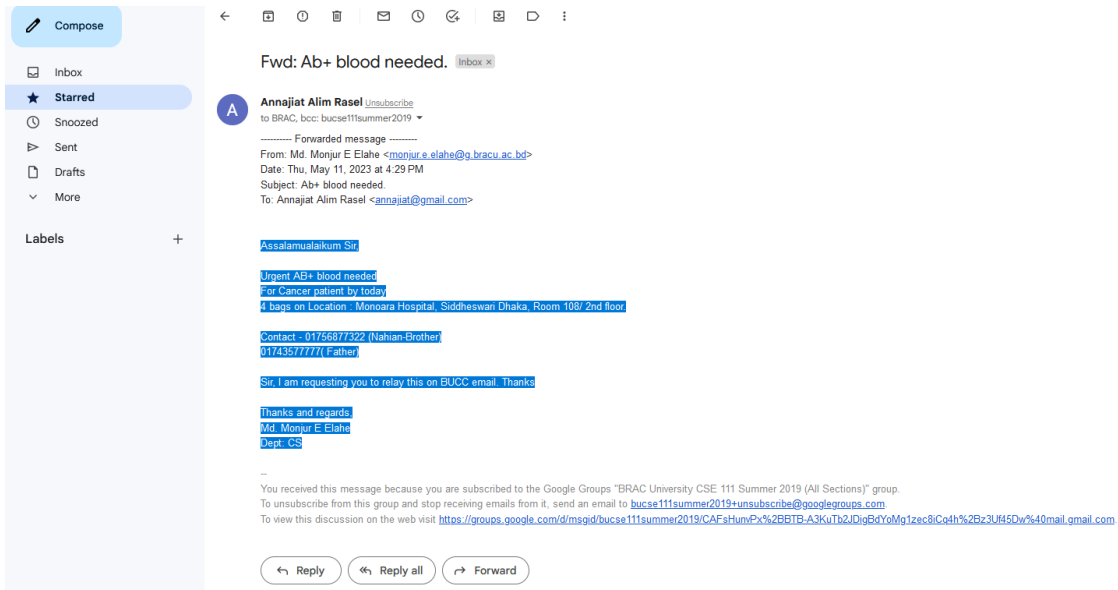


Figure 5.3: Getting the email

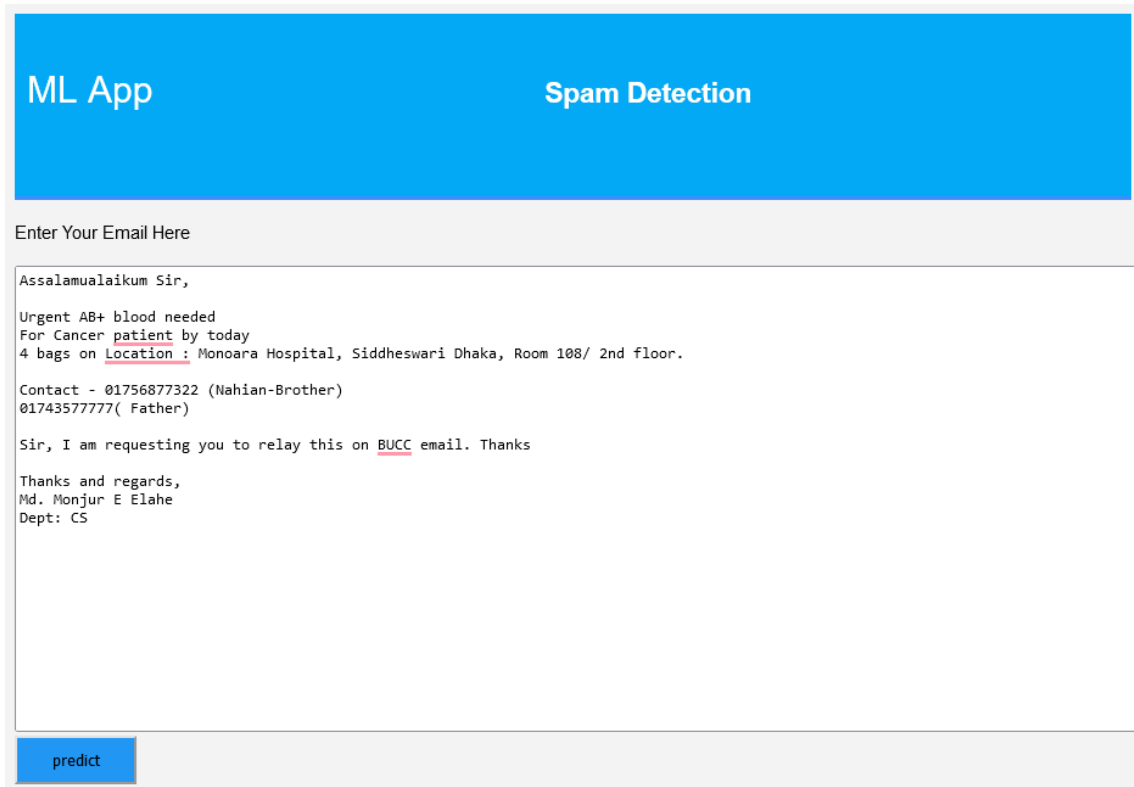


Figure 5.4: Entering the email

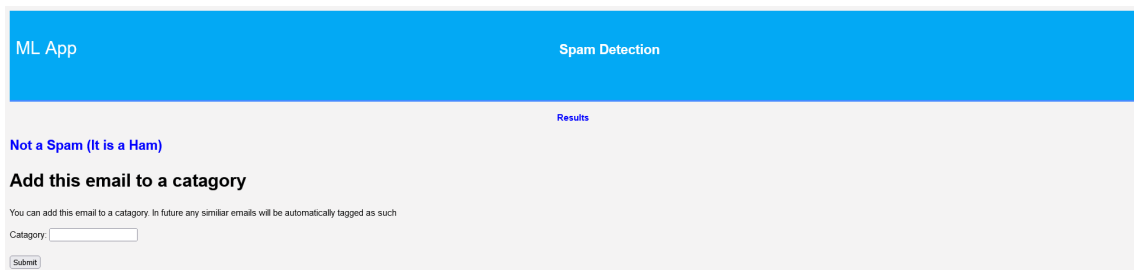


Figure 5.5: When it detects as Not Spam

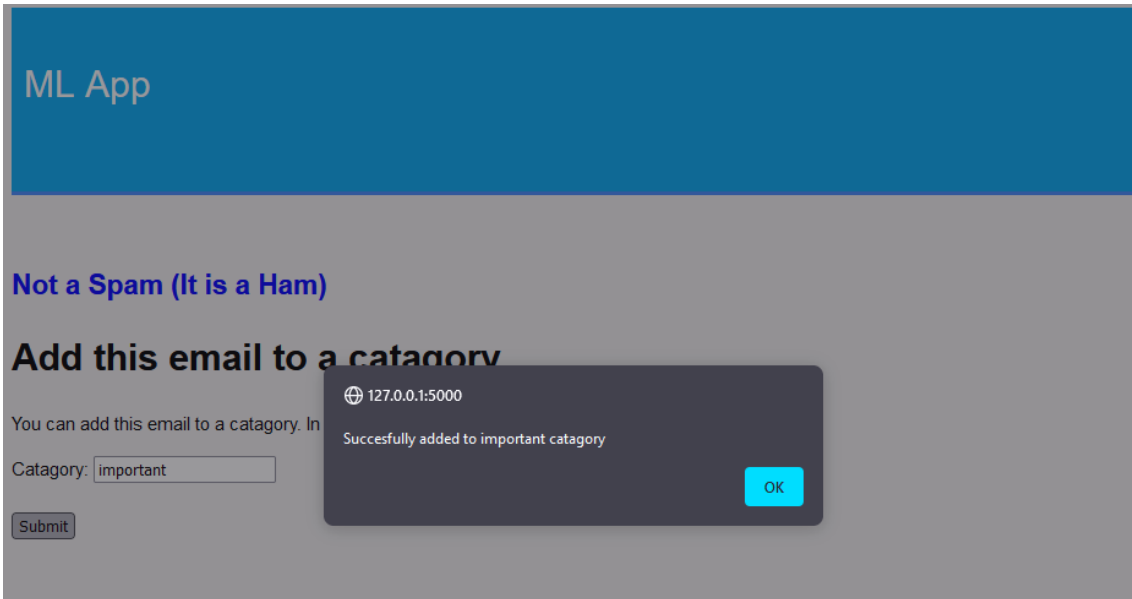


Figure 5.6: Adding the email to a category

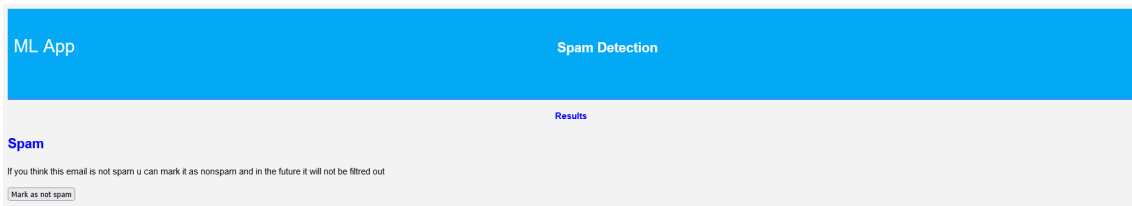


Figure 5.7: When it detects as Spam

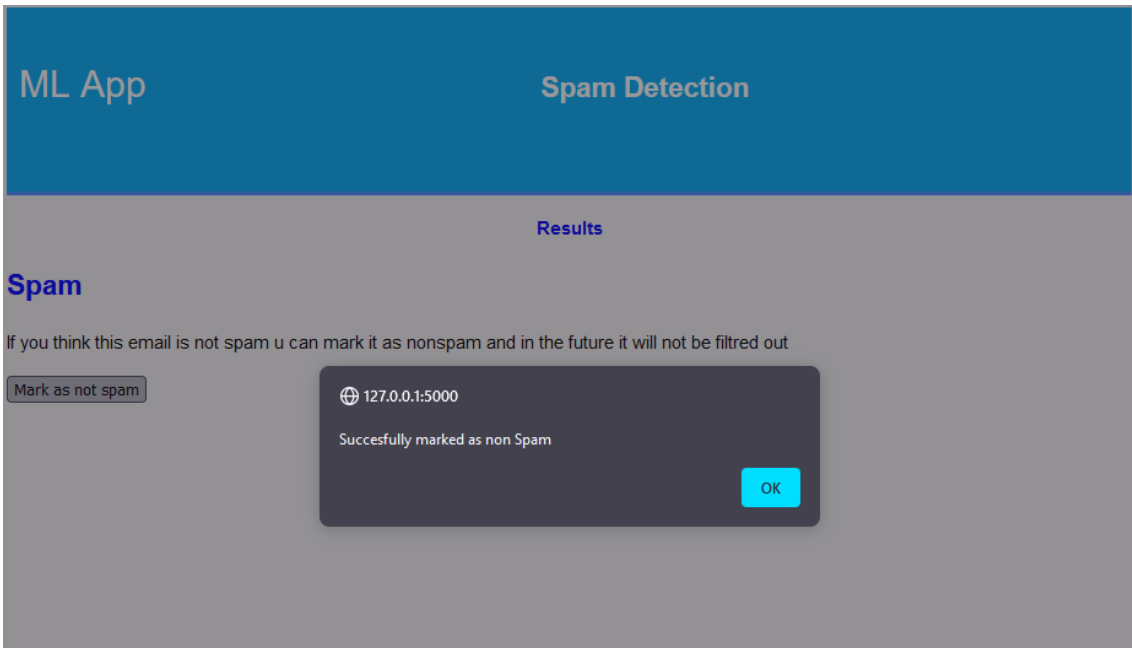


Figure 5.8: Marking a falsely detected Spam email

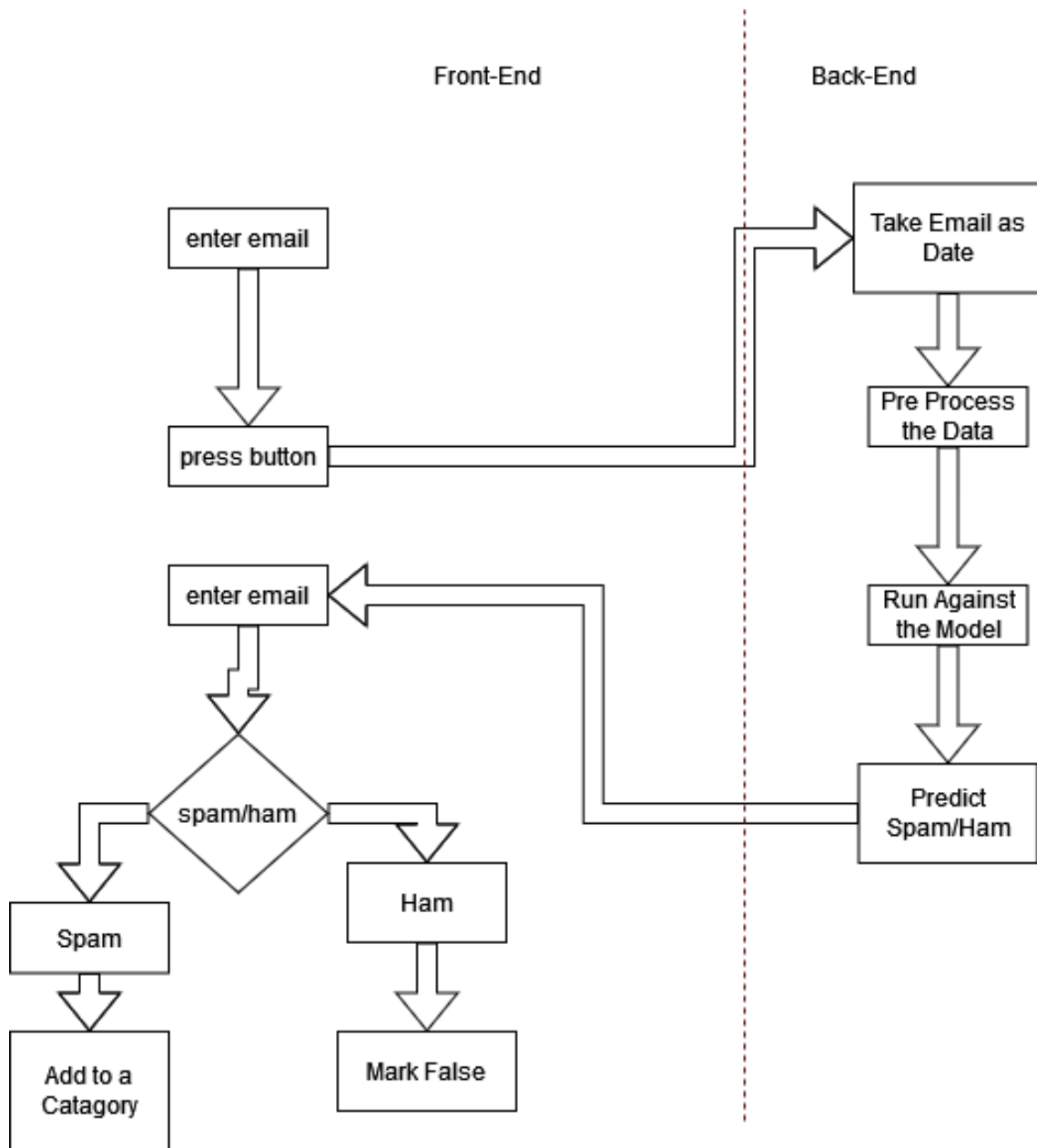


Figure 5.9: Web based Application

Chapter 6

Experimental Result

We have applied the Correlation Based Feature Selection (CFS) approach, in order to reduce dimensionality and find the most relevant features. Then we have select only relevant feature words from the data we need to apply a correlation-based feature selection approach.

$$CFS = \max_{sk} \left[\frac{r_{cf1} + r_{cf2} + \dots + r_{cfk}}{\sqrt{k + 2(r_{f1f2} + \dots + r_{fifj} + \dots + r_{fkfk-1})}} \right]$$

After applying CFS we created a heat map 6.1 :

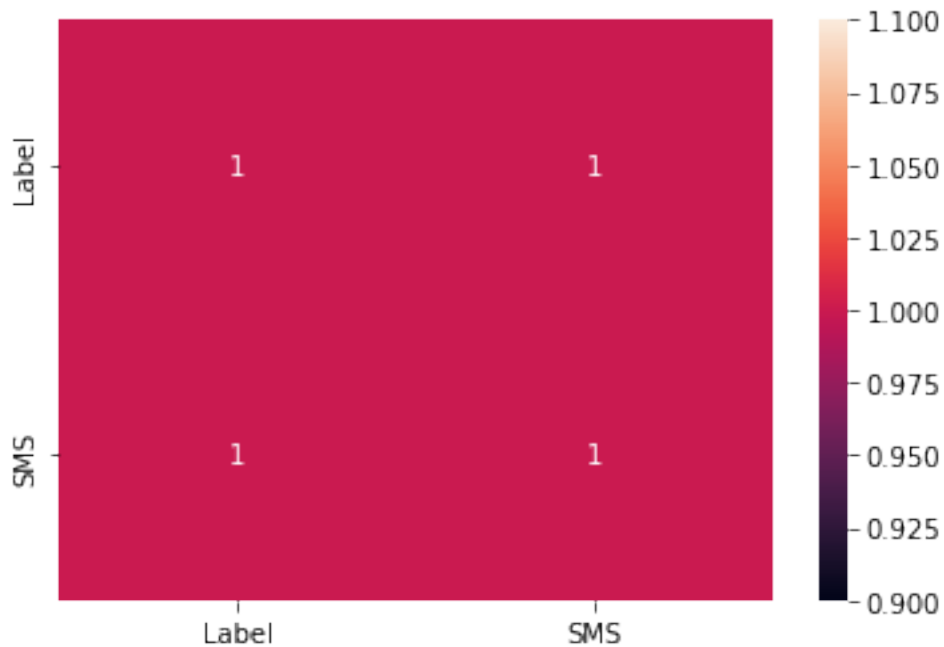


Figure 6.1: Heatmap of the model

At the end of our spam detection model, we tested the accuracy of our model. The accuracy percentage of our model is 97.666% as seen in figure 6.2.

As well as we have seen that as we keep on increasing the dataset and number of emails our accuracy score keeps on increasing.

```
[ ] 1 correct = 0
     2 total = test_set.shape[0]
     3
     4 for row in test_set.iterrows():
     5     row = row[1]
     6     if row['Label'] == row['predicted']:
     7         correct += 1
     8
     9 print('Correct:', correct)
    10 print('Incorrect:', total - correct)
    11 print('Accuracy:', correct/total)
```

```
Correct: 1088
Incorrect: 26
Accuracy: 0.9766606822262118
```

Figure 6.2: Accuracy of the model

We have successfully created a web application that tells us if the mail is spam or ham mail in an instant and allows us to categorize emails according to our own preferences from our inbox.

In accordance with N. Permar [14] we have calculated the precision, recall, f-measure, and accuracy of each algorithm with the assistance of True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). TP are the emails that are spam and correctly identified as one and TN are the ones not spam and correctly identified as such. Similarly, FP are non-spam emails that are identified as spam and FN are spam emails that are identified as non-spam.

Now, Precision, formulated as $P = \frac{TP}{TP+FP}$, gives the magnitude of proportion to TP and total positives. It is expected true positive spam email compared to the entire expected true positive observation.

Then, Recall, formulated as $R = \frac{TP}{TP+FN}$, is the ratio of TP spam email observed to actual email spam. Also, this gives how sensitive our model is.

Then, F-Measure formulated as $F = \frac{2PR}{TP+FN}$, is calculated from precision and recall.

Finally, accuracy, formulated as $A = \frac{TP+TN}{TP+TN+FP+FN}$, is all the positive result divided by all the email

	Precision	Recall	F-measure	Accuracy
NaiveBayes	96.5%	95.00%	96%	98%
SVM	97%	92%	95%	97.5%
LSTM	97%	95.4%	96.5%	98.4%
HMM	93.05%	89.95%	91.4%	95.9%
CNN	95.7%	90.6%	92%	96.2%
Our Model	96.7%	94%	95.2%	97.66%

Table 6.1: Performace of our model comparing with the existing models

After calculating the necessary parameters, we can see that NaiveBayes and SVM have a better performance compared to our model. However, if we increase our dataset our model provides better accuracy as it is constantly training itself with each new dataset provided thus our model is more suitable for scaling up with increasing data.

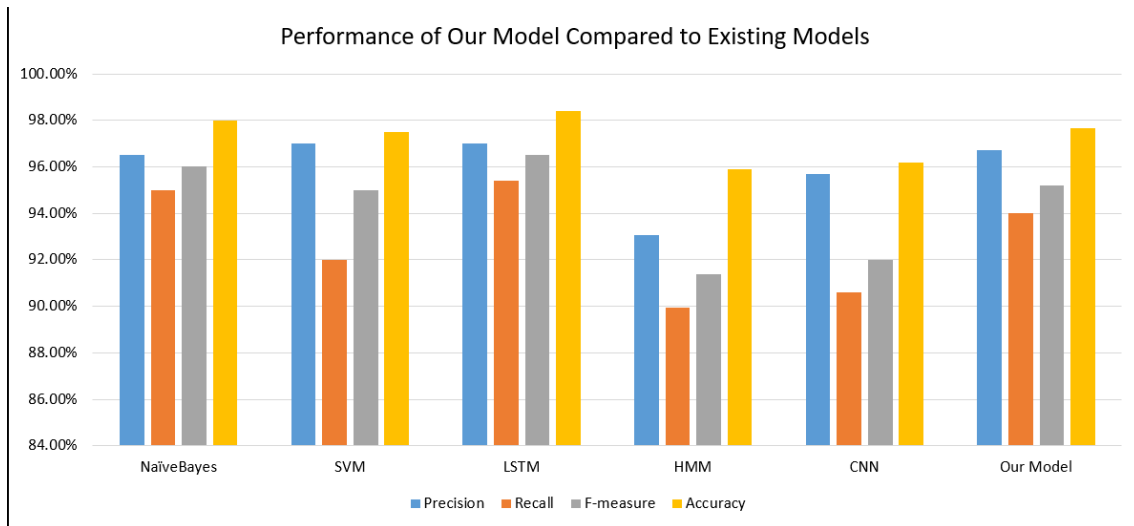


Figure 6.3: Graph comparison of our model with the existing models

Chapter 7

Research Contribution and Challenges

7.1 Contribution

7.1.1 Dataset Collection

We have created a primary dataset of more than 1700 spam emails. These spam emails were collected not only from BRAC University students but also from people from all spheres of life. It was not specific to any country; mail was collected from Bangladesh and international borders as well. Given the scarcity of spam mail data, we have given a sufficient number for future researchers to use. Our primary dataset contains distinct emails that cover a wide variety of sectors so that we could enrich the model further

7.1.2 Multiple AI Used

For the data preprocessing part of spam detection, we used CFS which is correlation-based feature selection, stemming, and lemmatization. Moreover, for spam detection, we used Naive Bayes.

7.1.3 Web Application and Categorization

Our proposed model created a web application where the user can import an email from its inbox and then the app tells us if the mail is spam or not. The web application also allows the user to categorize their email according to their own preference and makes their own categories depending on their incoming mail. It then becomes a powerful tool to use such models and systems in our daily lives. We really wanted to establish such a system so that the majority of the people can use it.

7.2 Challenges

7.2.1 Concern For Privacy

One of the main challenges that we faced is that at first people were scared to share their emails with us. There were concerns that people had if it is safe to share their spam emails. They were concerned about privacy as well as they feared that they might be hacked. Moreover, people were also concerned if the survey form is just another process of inundating their inboxes with dozens of spam emails.

7.2.2 Automatic Deletion of Spam Mail

Another major problem that we faced while collecting our primary dataset is that spam emails get deleted automatically from Gmail after 30 days. If it was not the case we could have created a more enriched primary dataset. However, because of this feature of Gmail and other email software our primary dataset is limited to more than 1700 spam emails.

7.2.3 Third Party Authorisation

At first, our aim was to create an extension. However, Gmail does not allow third-party access. Since we are the third party we did not get access due to which we were not able to extract emails and thus we could not create an extension. Because of this, we created a web application.

7.2.4 Integration of Naive Bayes and Particle Swarm Optimization for Spam Detection

Initially, our aim was to create a hybrid classifier using both Naive Bayes and Particle Swarm Optimization. However, we could not incorporate PSO in our model, as it works best with numerical values only, converging particles into one single outcome. And as our model dealt with tokens in the form of strings, it was difficult to apply PSO on words.

7.3 Shortcomings

The Particle Swarm Optimization (PSO) algorithm is used to optimize the outcome from a preceding algorithm. In our model, we initially aimed to create an integrated classifier which was the combined workings of the Naive Bayes and the PSO algorithms. However, we could not apply the aforementioned algorithms together because of some issues we faced in the PSO part of the model.

Particle Swarm Optimization works by initially taking a "swarm" or a group of "particles" which are meant to randomly move around in a search space with the goal of looking for the "food source". As our model will be used for emails, the particles will be the tokens that are extracted from the emails, and then these tokens will search for the food source, working in a similar manner as mentioned in [14]. This is where we faced the first problem. The tokens that were extracted from the email

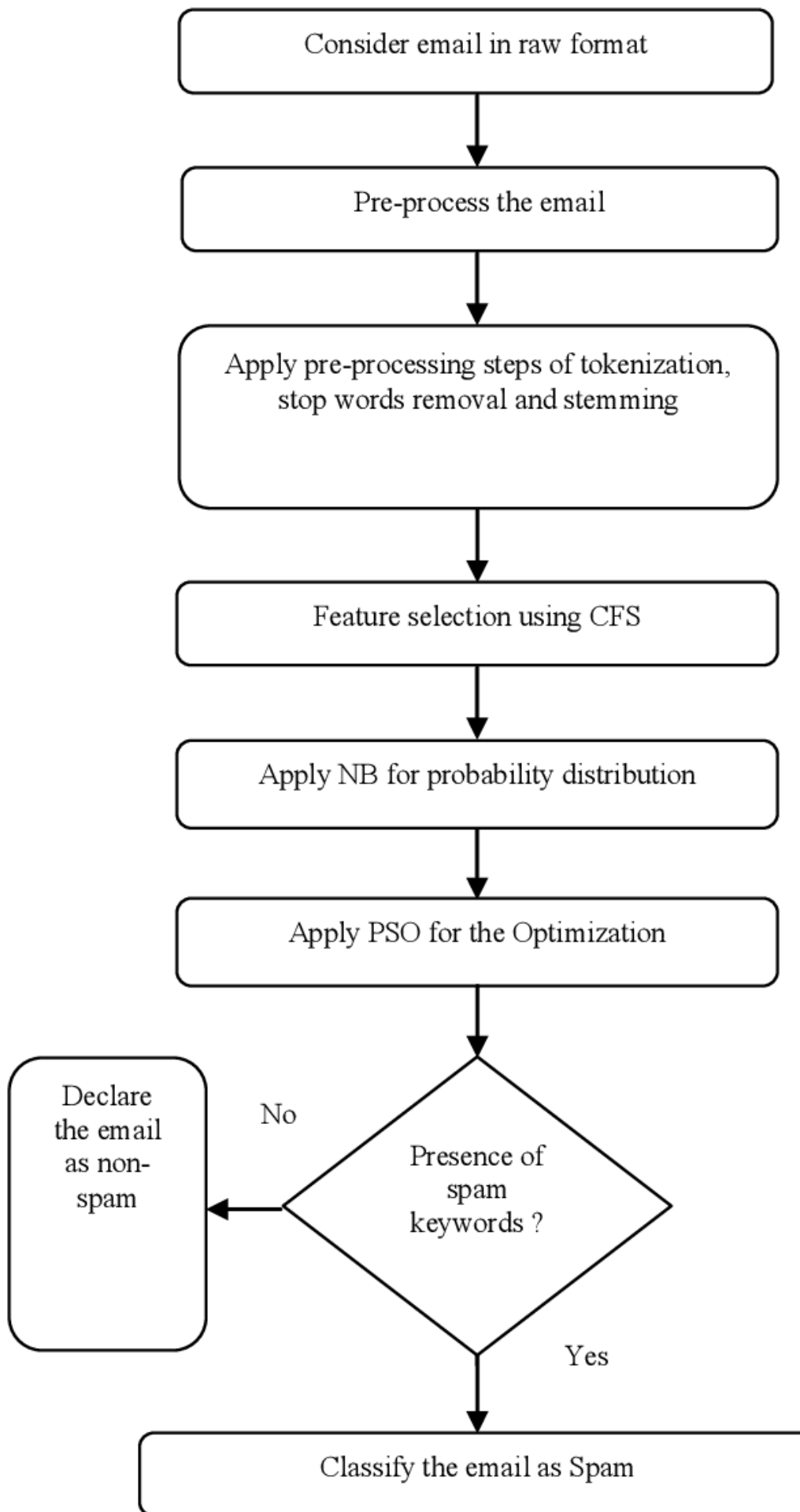


Figure 7.1: General Diagram of our Initial model

were words in their root or base forms. This meant that words or strings would have to be used to randomly move around in the search space.

The second problem we faced was regarding this search space. Generally, the search space is an n -dimensional space calibrated with numbers on the coordinate axes. Here we were having trouble making the tokens traverse through a numerical search space, even though the tokens moved around the space based on their current position (X_i) and randomly calculated velocity (V_i). The tokens' new positions were updated by adding the randomly generated velocity vectors to their respective current position vectors.

As to the third issue, we faced problems in fixing the food source for these particles or tokens. The particles are required to converge down into one single point, that is, the food source. Therefore, according to our goal, our food source was 'non-spam' words. But, there are a lot of non-spam words, and one word can never converge into another word if they are of different meanings. For example, if one of the tokens is the word "greetings", and it is not present in the previously attained non-spam words, that is the food source, then this token would not converge into any other words, even though we can see that it is a non-spam token.

Furthermore, Particle Swarm Optimization algorithms give a single value as output, that is, the optimized outcome only. But as our tokens are words or strings, and we cannot converge these strings into a single string or token, we found it quite difficult to apply PSO on string tokens to achieve an optimized output from all the particles in the initial swarm of tokens.

Alongside such discovery, we researched for better solutions to understand these issues further, only to find that PSO works best for numerical values. And as our model dealt with strings or words only, we could not apply the Particle Swarm Optimization to optimize the tokens fed into this algorithm from the Naive Bayes algorithm.

All-inclusive, these are the difficulties we encountered that led us to decide not to include the PSO algorithm as part of our model

Moreover, we wanted to create an extension. However, we were unable to create one because Gmail does not allow third-party authorization due to which we could not extract emails and hence instead of an extension we created a web application.

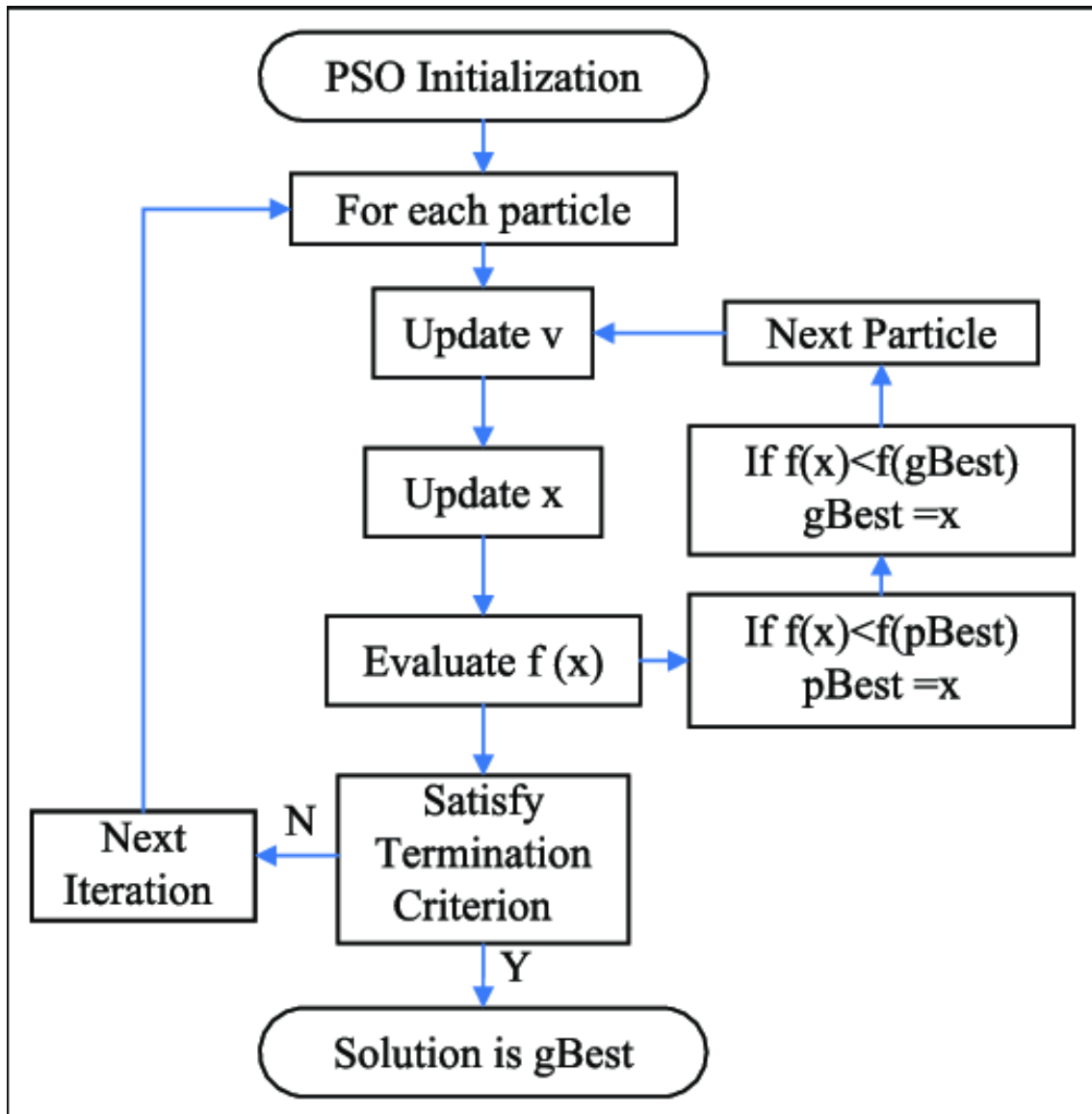


Figure 7.2: PSO [7]

Chapter 8

Future Work and Conclusion

8.1 Future Work

8.1.1 Better User Interface

Although we were able to establish a system, its user interface and design still to be desired. It needs improvement in order to make it easier, smoother, and frictionless for the user. An improved user interface will not only assist the user manage their email categories effectively, but it will also help the user keep track of changes and stored data.

8.1.2 Creating an extension

As we initially wanted to create an extension to implement our system with Gmail, we discovered that Gmail does not provide third-party authorization which led us to create a web app and manually read email content. However, we want to pursue this goal in the future to provide users with a seamless and smooth experience with our system, after receiving authorization from Gmail or by scraping emails to automate the process of reading emails. Using an extension will also allow the users to work on other apps simultaneously while cherishing the services that our system will provide.

8.1.3 Connecting Google Scholar With The Web Application

We aim to produce a student-friendly web application. Where due to certain mails which include certain research topics the web application may provide push notifications to the user about the research articles present on that topic on Google Scholar. This will aid students in their research and they would not miss any important updates regarding their academic interests

8.1.4 Connecting Google Calendar With The Web Application

In addition to the aforementioned future goals we want to achieve, we also want to incorporate the Google Calendar API into our system. This is to automate the scheduling of a meeting where required. Many times, such emails are sent where a

meeting is requested to be scheduled on a given date and time, and the user then has to manually set up the meeting event in his/her Google calendar. As this requires the attention and valuable time of the user, we aim to automate this process in order to make the user's activities more efficient and easy. Our probable work plan includes reading the email content to extract information like the date and time of a meeting requested by the email client. Then, the system will check the user's calendar to see if they are available at the requested time. If there is a free slot, a meeting will be scheduled and a prompt or a push notification will be sent to the user, confirming that a meeting has been scheduled. We think this will be very helpful for students, as well as office staff members, and many other people in different workplaces.

8.1.5 Detect and Filter Bengali Spam Content

Another feature we want to integrate into our system is detecting Bengali spam content. From our research, we have witnessed more and more emails are also sent in languages, other than English; and, we have encountered emails sent in the Bengali language. Additionally, we have forever been victims of numerous spam text messages on our mobile numbers. Therefore, to tackle these, we want to incorporate Bengali spam detection in our system to help everyone from this hassle.

8.1.6 Filter Social Media Spam Content

Furthermore, we want to adapt our system to more avenues so that it can detect and filter out spam text messages in social media applications like Facebook and Messenger, as from our conducted research, more and more people are getting spammed on their social media accounts. This will also help social media users to access such applications hassle-free and protect them and their accounts from inappropriate and unwanted content.

8.2 Conclusion

E-mails are an important part of our communication. This means keeping them safe, from countless nefarious and unwanted Emails which are trying to infiltrate users' inboxes, is key for better communication and safety of the user. With the help of machine learning, this work is striving to develop a robust system that can filter out these unwanted Emails. Moreover, this work explores the combination of three different works: the Naive Bayes for spam detection, personalized automatic email categorization, and web application for better and efficient results and to help find the right email in the right place.

Bibliography

- [1] T. Stephenson, “An introduction to bayesian network theory and usage,” Jan. 2000.
- [2] V. Christina, S. Karpagavalli, and G. Suganya, “A study on email spam filtering techniques,” *International Journal of Computer Applications*, vol. 12, no. 1, Dec. 2010. DOI: 10.5120/1645-2213.
- [3] S. Yoo, “Machine learning methods for personalized email prioritization,” Jan. 2010.
- [4] I. Idris and A. Selamat, “Improved email spam detection model with negative selection algorithm and particle swarm optimization,” *Applied Soft Computing*, vol. 22, pp. 11–27, Sep. 2014. DOI: 10.1016/j.asoc.2014.05.002.
- [5] J. Alqatawna, H. Faris, K. Jaradat, M. Al-Zewairi, and O. Adwan, “Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution,” *International Journal of Communications, Network and System Sciences*, vol. 08, no. 05, pp. 118–129, 2015. DOI: 10.4236/ijcns.2015.85014.
- [6] S. K. Trivedi, “A study of machine learning classifiers for spam detection,” *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 176–180, 2016. DOI: 10.1109/ISCBI.2016.7743279.
- [7] S. Wang, J. Yang, G. Liu, S. Du, and J. Yan, “Multi-objective path finding in stochastic networks using a biogeography-based optimization method,” *SIMULATION*, vol. 92, Jan. 2016. DOI: 10.1177/0037549715623847.
- [8] W. Hijawi, H. Faris, J. Alqatawna, I. Aljarah, A. Al-Zoubi, and M. Habib, “Emfet: E-mail features extraction tool,” Nov. 2017. DOI: 10.13140/RG.2.2.32995.45603.
- [9] K. Agarwal and T. Kumar, “Email spam detection using integrated approach of naïve bayes and particle swarm optimization,” pp. 685–690, Jun. 2018. DOI: 10.1109/ICCONS.2018.8662957.
- [10] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: Review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, 2019. DOI: 10.1016/j.heliyon.2019.e01802.
- [11] S. Douzi, F. Alshahwan, M. Lemoudden, and B. Ouahidi, “Hybrid email spam detection model using artificial intelligence,” *International Journal of Machine Learning and Computing*, vol. 10, pp. 316–322, Feb. 2020. DOI: 10.18178/ijmlc.2020.10.2.937.

- [12] S. Gibson, B. Issac, L. Zhang, and S. Jacob, “Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms,” *IEEE Access*, vol. 8, pp. 187 914–187 932, Jan. 2020. DOI: 10.1109/ACCESS.2020.3030751.
- [13] M. Izatt, *How gmail sorts your email based on your preferences*, Jan. 2020. [Online]. Available: <https://cloud.google.com/blog/products/gmail/how-gmail-sorts-your-email-based-on-your-preferences>.
- [14] N. Parmar, A. Sharma, H. Jain, and A. Kadam, “Email spam detection using naïve bayes and particle swarm optimization,” vol. Volume 6, pp. 367–373, Mar. 2020.
- [15] T. Xia and X. Chen, “A discrete hidden markov model for sms spam detection,” *Applied Sciences*, vol. 10, no. 14, p. 5011, 2020. DOI: 10.3390/app10145011.
- [16] N. Donges, *Random forest algorithm: A complete guide*, Jul. 2021. [Online]. Available: <https://builtin.com/data-science/random-forest-algorithm>.
- [17] M. RAZA, N. D. Jayasinghe, and M. M. A. Muslam, “A comprehensive review on email spam classification using machine learning algorithms,” *2021 International Conference on Information Networking (ICOIN)*, pp. 327–332, 2021. DOI: 10.1109/ICOIN50884.2021.9334020.
- [18] Z. B. Siddique, M. A. Khan, I. U. Din, A. Almogren, I. Mohiuddin, and S. Nazir, “Machine learning-based detection of spam emails,” *Scientific Programming*, vol. 2021, pp. 1–11, 2021. DOI: 10.1155/2021/6508784.
- [19] T. Kulikova and T. Shcherbakova, *Kaspersky spam and phishing report for 2021*, Feb. 2022. [Online]. Available: <https://securelist.com/spam-and-phishing-in-2021/105713/#share-of-spam-in-mail-traffic>.
- [20] I. LaBianca, *How spam filters work (and how to stop emails going to spam)*, May 2022. [Online]. Available: <https://www.theseventhssense.com/blog/how-spam-filters-work-and-how-to-stop-emails-going-to-spam>.