Bachelor of Science in Computer Science & Engineering

# Content-based Spam Email Detection Using N-gram Machine Learning Approach

by

Nusrat Jahan Euna

ID: 1504094

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

# Content-based Spam Email Detection Using N-gram Machine Learning Approach

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Nusrat Jahan Euna
ID: 1504094

Supervised by

Dr.Md.Iqbal Hasan Sarker
Assistant Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Content-based Spam Email Detection Using N-gram Machine Learning Approach'** submitted by ID: 1504094, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

## Board of Examiners

Chairman

Dr.Md.Iqbal Hasan Sarker
Assistant Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Asaduzzaman
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

# Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Nusrat Jahan Euna

**Signature of the candidate**

**Date: 19 April, 2021**

# Acknowledgements

The process of writing this thesis to completion has only been possible thanks to the help and supervision of a large number of people. Both personally and professionally, this has been a rewarding experience.

I'd like to express my deepest gratitude to my supervisor, Dr. Md.Iqbal Hasan Sarker, for guiding me through this study. I am grateful for his constant advice and probing questions, which have pushed me to think outside of my comfort zone and given me the confidence to take on challenges I never felt I could handle.

My parents, brother, and sister deserve praise for their patience in bearing with me during this ordeal. Their unwavering guidance and motivation have helped me in every area of my life.

# Abstract

E-mail is a cost-effective method of communication because it saves time and money. As a result, it is a popular method of personal and professional communication. E-mails enable internet users to quickly send and receive information around the world. However, there is a chance that your e-mails will be harmed by active or passive attacks. We sometimes receive e-mail from unknown sources, as well as e-mail containing irrelevant information.

Spam Mails are the term for these types of unwanted emails. Spam email is the practice of sending vast amounts of unnecessary data or bulk data to specific email addresses on a regular basis. Spam Mail is a form of electronic spam that involves sending nearly identical messages to different people via email. Malware can also be found in spam emails as scripts or other executable file attachments.

Based on the principle of content-based filtering, I have proposed a novel approach for classifying spam and ham emails in this project. A content-based filter parses messages' content, looking for terms that are often used in spam emails. To build the system various steps like preprocessing, feature extraction , model training have been performed. preprocessing steps are performed to make the raw data in a understandable form. To extract features from the preprocessed data we used n-gram and word2vec feature extraction method. We have also performed word-ngram character-n gram and combination of variable length n-gram.Then we have trained the model. The system's efficiency was then assessed using a variety of machine learning algorithms.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction

When the number of internet users grows, so does the use of electronic email by users as part of their everyday lives. As more and more people subscribe to various websites, goods, programs, catalogs, newsletters, and other forms of electronic communication, users are more likely to receive irritating spam messages and malicious phishing messages.

The so-called "spam" is one of the most serious issues that all e-mail users face. Spam is a term used to describe unsolicited commercial e-mail that is sent in bulk to thousands or even millions of people. Spam may be considered by some to be e-mail created by mass-mailing viruses or Trojan horses

Spam messages waste both valuable time of the users and important bandwidth of internet connections. Moreover, they are usually associated with annoying material or the distribution of computer viruses.As a results of the massive number of spam emails being sent across the net day after day, most email providers offer a spam filter that automatically flags likely spam messages and separates them from the ham.

Various filters use several techniques, most rely heavily on the analysis of the contents of an email via text analytic. Hence, there is an increasing need for efficient anti-spam filters that either automate the detection and removal of spam messages or inform the user of potential spam messages.

In this project ,we will learn how word-gram, character-gram and combination of variable length n-gram works. Then we will analyze their performance using various machine learning techniques and also classify spam and ham emails.

## 1.2 Framework for content based spam filtering

The problem of spam email has become a major problem now-a-days.Many researches have be done to classify spam and ham emails correctly.There are various techniques of classifying emails. In this project i have proposed a novel method for classifying spam and ham emails based on the concept of content based filtering. A content based filtering parses the content of messages, scanning for words that are commonly used in spam emails.Then various machine learning algorithms have been applied to evaluate the system performance. Though some steps have been followed to implement the system:

1.preprocessing steps are performed to transform raw data into a understandable form in machine learning.

2.Feature extraction methods are performed to get useful features from the existing data.Here we have used n-gram  word2vec methods to extract features.

3.Features that we obtained from the previous step were used to train the machine learning model by employing different popular classification algorithms.

4.After successfully training up the machine, we have to check the system before declaring it to be a content based spam filtering machine.

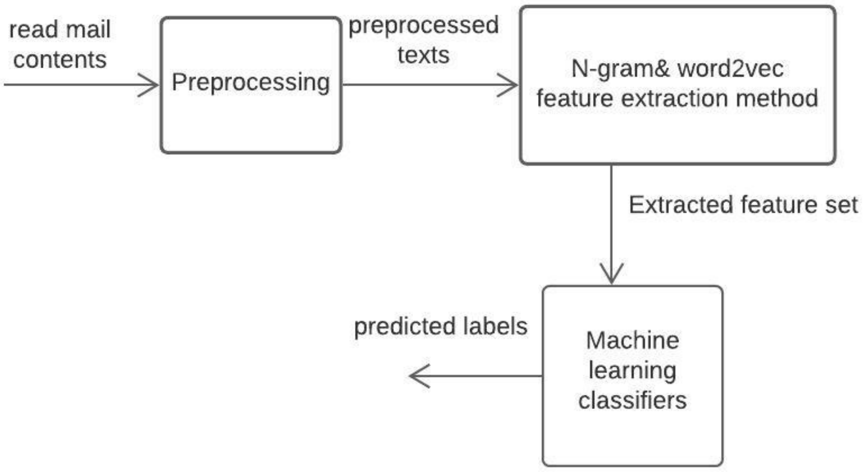


Figure 1.1: block diagram for a content based spam filtering

## 1.3 Difficulties

No spam filter can detect spam with 100 percent accuracy. So, we have also faced some difficulties during the implementation of the system.As it is a content based spam filter so it is important to extract the useful features from the mail content. Again to extract the features it was also necessary to preprocess the text very well. As noisy text can decrease the performance of the system.The major problems that have faced during implementation are given below:

1.Extracting the useful feature from the text is quite difficult as there can be various features.

2. Training the model the is also challenging because if we don't train the model properly it could provide wrong result.

3. lowering the false positive rate with appropriate analysis and getting a comparative analysis with various machine learning algorithm and getting better accuracy was also matter of concern.

## 1.4 Applications

Applications of spam filtering system:

- Without spam filtering, a business email system is extremely vulnerable, if not unusable. To secure a network from the many potential threats, such as viruses, phishing attacks, broken web links, and other malicious material, it's critical to avoid as much spam as possible.
- filters also prevent the servers from being overburdened with non-essential emails, as well as the more serious issue of being compromised with spam software, which could transform them into spam servers.
- filters provide an extra layer of security for a customer, a network, and a company by stopping spam email from reaching their inboxes.

## 1.5 Motivation

Spam filtering is a must in today's world if we want to keep our company secure. It is estimated that world wide email sent is 144.8 billion messages per day and 65 percent of it is spam and the amount of spam continues to rise as spam remains a profitable industry. Spammers' strategies to get their messages into our inboxes and destroy our system. Spam filtering software must be modified on a regular basis to keep up with the threat.

Despite the fact that people have the ability to identify spam emails quickly, doing so can be time consuming. A machine learning approach is used to simplify a computer's classification task in an automated manner. Constrained data and text written in an informal manner are the most likely problems that caused the current algorithms to fail to meet the demands during classification due to a lack of data sets for email spam.[1]

To overcome these problems we will establish a content based spam filtering performing combination of variable length n-gram. we will also perform word-gram, character-gram and word2vec.And also perform various classification algorithms to analyze their performance and get better accuracy.

## 1.6 Contribution of the thesis

Thesis or Research work is performed, to achieve a specific set of goals, whether it is to define a new methodology or to improve the existing ones. In this thesis, the main focus was given to improve the accuracy of the spam email detecting methods. The primary contribution of this thesis is the following:

- Performing combination of variable length n-gram
- Performance evaluation of different feature extraction techniques.
- Incorporating different machine learning algorithms on the extracted features.

## 1.7 Thesis Organization

The rest of this thesis report is organized as follows:

- Chapter 2 gives a brief summary of previous research works in the field of content based spam filtering.
- Chapter 3 describes the proposed methodology for the content based spam filtering. In the proposed framework, feature extraction is performed by two different methods : n-gram  word2vec. where word-gram,character-gram and combination of variable length n-gram also performed. For classification purpose various machine learning classifier have been used.
- Chapter 4 provides the description of the working dataset and analysis of the performance measure for the proposed framework.
- Chapter 5 contains the overall summary of this thesis work and provides some future recommendations as well.

## 1.8 Conclusion

In this chapter, an overview of of content based spam filtering has been provided. Along with the difficulties, the summary of the spam email filtering method framework is described in this chapter. The motivation behind this work and contributions are also stated here. In the next chapter, background and present state of the problem will be provided.

# Chapter 2

# Literature Review

## 2.1 Introduction

Spam emails have been a big issue in recent years. Eminent scholars have put in a lot of work to classify emails into categories.The spam/ham group, as well as the detection rate for each. They're Machine learning was used to complete their tasks. To The writers have also carried out the research to solve these problems.classification task with the help of a modern spam filter technique and the rate at which they can be detected. In this section we will discuss work carried out by a number of distinguished scholars together with the dataset used, the tools used, and the extraction with characteristics and outcomes.

## 2.2 Related Literature Review

Liu and T. Moh proposes a spam email filtering method based on content. The proposed algorithm is divided into two phases: training and classification. Individual users' emails are collected from training datasets during the training process. Following the collection of email material, the next move was to build a spam and ham keywords corpus, which was compared to the keywords extracted from individual users' emails. The corpus is built using a threshold value. As a result, the method is inefficient. The system's overall accuracy is 92.8 percent. [1]

Gaurav et.al based on the concept of, suggested a spam mail detection (SPMD) system for detecting spam emails. For classification, three algorithms are used: Naive Bayes, Decision Tree, and Random Forest. Random Forest was found to have the highest accuracy of 92.97 percent among these classifiers.[2]

Ioannis Kanar and colleagues proposed a spam detection method based on low-level data. Instead of the standard 'bag of words' representation, they used character ngram to build a 'bag of character n-grams' representation. where the variable length was determined in advance [3]

Kiliroor et al. suggested a model for detecting unwanted or unsolicited messages on user walls on online social networking sites. In addition to a standard Bayesian classifier, the proposed method uses SLC features to measure the sender-receiver relationship.[4]

To detect spam emails, Weimiao Feng et al. proposed an SVM-NB method. When the NB algorithm is used, SVM-NB aims to remove the presumption of independence among features extracted from the training set. The SVM technique is used to classify training samples into various groups and to identify dependent training samples.When those samples are removed, the training set becomes more autonomous, with less overlapping features.[5]

Rathi et al. suggested using different classifiers to detect spam emails. They used the entire data set and applied the algorithm one by one, without picking a feature. They detect spam mails in the second part by applying feature selection algorithms first, the algorithm they used here is Best-First Feature Selection algorithm, and then applying all the classifiers one by one on the reduced data set with selected features and checking the output.[6]

The learning-based spam filter family includes Naive Bayes. The algorithm was created by Paul in 2002. Paul Graham's previous formula, which works better than any spam filter, was published in 2004. Graham demonstrated that the pantel and Lins filters only detect 92 percent of spam.[7]

The researcher proposed a spam mail filtering system based on n-gram indexing and help vector machines in this paper (SVM). They practiced with e-mails obtained from various users and performed the filtering procedure with SVM classifier, which uses index term created by n-gram indexing and data set generated by word dictionary, for the experiment of proposed methods. As SVM kernel functions, they used dot, polynomial, and radial functions.[8]

Spam detection using N-gram analysis and machine learning techniques is proposed in this paper. They have only removed stop words that are common in all other emails, leaving only those words that are unique to spam and the ham domain, which can provide much more detail to the classifiers during the training stage.[9]

In this paper, the Naive Bayes classifier was used in the filtering of e-mail spam. The efficiency of the Naive Bayes classifier is also determined by the datasets used.[10]

This paper compares the efficiency of various supervised machine learning algorithms for detecting spam and ham messages, such as the nave Bayes Algorithm, help vector machines algorithm, and maximum entropy algorithm, in filtering Ham and Spam messages.[11]

Khamis S.A proposed a frame work working with email header features for email spam detection by analyzing two (2) email datasets: Anomaly Detection Challenges and Cyber Security Data Mining from websites. The main goal of this study was to extract the appropriate features of the email header from the datasets and test them to identify the features using Support Vector Machine (SVM) using RapidMiner Studio and Weka 3.9.2. Data Collection, Data Pre-Processing, Features Selection, Classification, and Detection are the five (5) phases of the technique. The Support Vector Machine (SVM) was used to classify email which provides 80 percent of accuracy.[12]

This paper proposes two widely used machine learning approaches, the Nave Bayes Classifier and the Support Vector Machine, for classifying emails as spam or ham based on their body or material. Independent terms are called features in the Nave Bayes Classifier. The Support Vector Machine can be used to represent an email in vector space, with each function corresponding to a single dimension. Finally, accuracy, recall, and F-measure output metrics are compared between two approaches.[13]

In this paper , the artificial bee colony algorithm is combined with a logistic regression classification model to create a spam detection system. The findings were based on three publicly accessible datasets (Enron, CSDMC2010, and TurkishEmail). They also compared the proposed model's spam detection efficiency to that of support vector machines, logistic regression, and naive Bayes classifiers, as well as the performance of previous studies' state-of-the-art methods.[14]

In this paper, the study of spam filtering employs the relief feature selection technique to select features and then classifies data using a fuzzy support vector machine to deal with factors of unknown type.[15]

S. Abiramasundari et.al proposed a technique of Rule Based Subject Analysis (RBSA) and Semantic Based Feature Selection (SBFS) which is combined with Machine Learning algorithms . To check the subject field of emails, a number of rules have been created. To minimize the features, a semantic-based feature selection technique is applied to the email content. Support Vector Machine, Multinomial Naive Bayes, Gaussian Naive Bayes, and Bernoulli Naive Bayes are four classifiers built into RBSA and SBFS. On the Enron dataset, the efficacy of the suggested techniques is evaluated.[16]

In this paper, features are divided into five classes in the proposed method: user profile features, account details features, user activity based features, user interaction based features, and tweet content-based features, with a total of 28 features. An optimal subset of these features is chosen for the learning process in the feature selection stage. Two Gaussian and polynomial kernels, on the other hand, use a support vector classifier for the learning process. Finally, standard parameters are used to equate the proposed approach to multi-layer perceptron (MLP), Naive Bayes (NB), random forest (RF), and k-nearest neighbors (KNN) approaches.[17]

RakeshNayak et.al , proposed approach of data science for spam email detection (SMD) using machine learning algorithm. This proposed method for spam email detection employs a hybrid bagging approach that combines the Nave Bayes and J48 (i.e. decision tree) machine learning algorithms. Each of these algorithms is run on a data set that has been partitioned into different sets using data science.[18]

This paper conducts an in-depth examination of spam information in order to propose an effective spam filtering system that can adapt to a changing world. They concentrated on email header analysis and use a decision tree data mining technique to search for spam association guidelines. Then, based on these association laws, they proposed an effective systematic filtering process.[19]

In this paper , proposed a new spam filter that combines an N-gram tf.idf feature selection, a modified distribution-based balancing algorithm, and a regularized deep multi-layer perceptron NN model with rectified linear units with an N-gram tf.idf feature selection, a modified distribution-based balancing algorithm, and a regularized deep multi-layer perceptron NN model with rectified linear units.[20]

In this paper, they introduced an effective spam filtering method based on decision tree data mining technique , analyzed spam association rules, and applied these rules to create a systematized spam filtering. method.[21]

In this paper for classifying spam and non spam emails, they proposed an intelligent technique based on Naive Bayesian classification that uses custom data sets in conjunction with existing datasets.[22]

This paper presented a novel feature-selection approach that uses semantic information to detect topics and shows how it can be applied to spam filtering. We contrasted the results obtained with our proposal to those obtained with Information Gain (the most widely used feature selection method in the spam filtering domain) and Latent Dirichlet Allocation (the most widely used feature selection method in the spam filtering domain) (a well known unsupervised technique in the text-mining domain). When the novel approach was used, the performance of the tested classifiers improved significantly. These findings support the idea of filtering spam by using topics rather than words (tokens). Furthermore, the latest feature selection approach will automatically discard features.[23]

This paper takes a different approach by using rules to verify the header and URL, as well as analyzing the body text.Bayesian classifier and Apriori algorithm are used to classify files and attachments.[24]

In this paper, they created a spam filter that analyzes the header, URL, body, and attachments to distinguish spam from non-spam (ham) emails. The header and URL are validated against rules, and the body text and attachments are validated using a Bayesian classifier and the Apriori algorithm. Only attachment files (*.rtf, *.txt, *.docx, *.doc,*.pdf) are tested.[25]

## 2.3 Conclusion

In this chapter, a detailed literature review is discussed. There are various approach of spam filtering. we have briefly discussed about the previous methods some of their limitations and analyzed their performance. Different feature extraction techniques and classifiers used by the researchers are also described here. The next chapter contains the detailed explanation of the proposed methodology of spam email detection framework.

### 2.3.1 Implementation Challenges

As spammers adopting new techniques day-by-day so it is difficult to detect the spam emails and legitimate emails. Spammers use simple trick to avoid filters by misspelling keywords introducing valid text line like Bible versus into spam messages,using various HTML techniques to trick filters.

Again spammer use Botnets which consists of "zoombie" computers both home and corporate personal computers that are infected with a worm, virus entity or trojan horse that allows them to be controlled by a spammer. the advantages to spammers of using botnets are that they can avoid detection by Internet service provider.

Now-a-days spammers also sending image based spam , audio, videos etc. Spam with attachments : using PDF files, Excel worksheets etc are using to carry the spam content.

So considering above criteria it is quite difficult to make a spam filter which could classify spam emails accurately.

# Chapter 3

# Methodology

## 3.1 Introduction

Spam filters work by analyzing our emails before they reach our inbox to determine if they are spam or not. They analyze the content, email address, header, attachments, and language of your emails, and scans them for anything "suspicious".However , in our proposed methodology we are working with the contents of the emails we have used two different feature extraction methods : n-gram and word2vec. where word-gram,character-gram and combination of variable length n-gram also performed and various classification methods have been used to classify emails and evaluate the system performance.

## 3.2 Steps of proposed methodology:

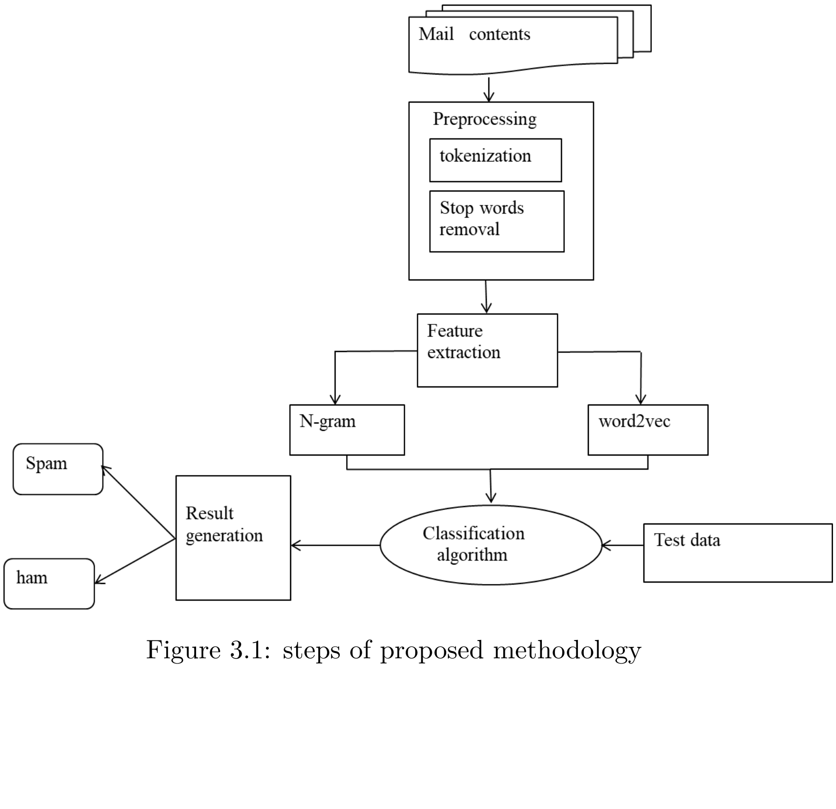


Figure 3.1: steps of proposed methodology

## 3.3 Detailed Explanation

The main goal of this project is to create a machine learning-based framework that can classify SPAM and HAM emails based on their contents. Figure 3.1 depicts the proposed system's schematic method, which is divided into four parts: preprocessing, feature extraction, training, and prediction. Several preprocessing steps are followed before the input texts are processed. To extract features from the processed documents, feature extraction methods are used. Machine learning classifiers (such as SVM, Logistic regression, Decision tree, Random forest, and Multinomial Nave Bayes) are trained using exploited features in the training process. Finally, the model that has been trained will be used for classification in the prediction step.

### 3.3.1 preprocessing

Data preprocessing is an essential step in Machine Learning. Preprocessing is used to transform raw data into a natural form by removing inconsistencies and inaccuracies. therefore, we should preprocess our data before feeding it into our model. Suppose an email :("hello! we would like to offer software localized version. . . .") .This email can be preprocessed as follows:

- Redundant character removal: Special characters like(<,=,>), numbers, and punctuations are removed from each text. After this, the text would become( "hello we would like to offer software localized version").
- Removal of stop words: Words like "the", "an", "this", "a", etc. which are not necessary for identifying spam or ham emails. while tokenizing we won't count the words by using the stop word list. After removing the stop word the text becomes-("would like to offer software localized version").
- Tokenization: Tokenization is the process of splitting the given text into smaller pieces called tokens. Tokenization gives a list of words like-("would", "like", "to", "offer", "software", "localized", "version").
- Lemmatizing the word: Lemmatization normally leads to doing things accurately with the use of a vocabulary and morphological analysis of words,

trying to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. After lemmatizing the text we get-("would", "like", "to", "offer", "software", "localize", "version").

### 3.3.2 Feature extraction

Feature extraction is the process of getting useful features from existing data. It builds relevant information from raw data to features – by reformatting, merging, converting primary features into new ones until it yields a new set of data that can be utilized by the machine learning models to achieve their goals. We have extracted two features from the texts-

1.N-gram- In n-gram we have also performed –word n-gram, character n-gram and combination of variable length n-gram

2.Word2vec

### 3.3.3 N-gram:

An n-gram is a n-tuple or group of n words or characters (grams, for pieces of grammar) which follow one another. Here, 'n' indicates the number of consecutive words that can be treated as one gram. The model extracted linguistic n-gram features of the texts. The N-gram approach is used to take into account the sequence order in a sentence in order to make more sense from the sentences. In our model we have worked with both word-gram, character-gram and combination of variable length n-gram.

### 3.3.4 Word n-gram:

In word n-gram ,we work with tuple or group words. Table-3.1 shows the word-gram representation of an email.

### 3.3.5 Character n-gram:

A character n-gram is a text that is represented by a series of characters. Character n-grams, in contrast to word n-grams, which can only detect the identity

| Sentence | “we would like to offer software localized version” |
|---|---|
| Uni-gram | ‘we’, ‘would’, ‘like’, ‘to’, ‘offer’, ‘software’, ‘localized’, ‘version’, |
| Bi-gram | ‘we would’, ‘would like’, ‘like to’, ‘to offer’, ‘offer software’, ‘software localized’, ‘localized version’ |
| Tri-gram | ‘we would like’, ‘ would like to’, ‘like to offer’, ‘to offer software ’, ‘offer software localized’, ‘software localized version’ |

Table 3.1: word n-gram representation of an email

of a word and its potential neighbors, can also detect the morphological makeup of a word. Character n-grams are particularly effective at identifying trends in misspellings in tasks like NLI, where several words are likely to be misspelled. They are also significantly less sparse than word n-grams. The character gram representation of an email is shown in Table 3.2.

| Sentence | “we would like to offer software localized version” |
|---|---|
| Uni-gram | ‘w’, ‘e’, ‘w’, ‘o’, ‘u’, ‘l’, ‘d’, ‘l’, ‘i’, ‘k’, ‘e’, ‘t’, ‘o’, ‘o’, ‘f’, ‘f”, ‘e’, ‘r’, |
| Bi-gram | ‘we’, ‘ew’, ‘wo’, ‘ou’, ‘ld’, ‘li’, ‘ik’, ‘ke’, ‘et’, ‘to’, ‘of’, ‘fe’, ‘er’ |
| Tri-gram | ‘wew’, ‘ewo’, ‘wou’, ‘oul’, ‘uld’, ‘ldi’, ‘dli’, ‘ike’, ‘ket’, ‘eto’, ‘too’, ‘off’, ‘ffe’, ‘fer’, |

Table 3.2: character n-gram representation of an email

### 3.3.6 Combination of variable length N-gram:

In combination of variable length we don’t pre-define the variable length. Here we can combine 1-gram with 2-gram or 3-gram with 5-gram. Table-3.3 shows the word-gram representation of an email.

| Sentence | “we would like to offer software localized version” |
|---|---|
| 1,2-gram | ‘we’, ‘would’, ‘like’, ‘to’, ‘offer’, ‘software’, ‘localized’, ‘version’, ‘we would’, ‘would like’,<br>‘like to’, ‘to offer’, ‘offer software’, ‘software localized’, ‘localized version’ |
| 2,3-gram | ‘we would’, ‘would like’, ‘like to’, ‘to offer’, ‘offer software’, ‘software localized’, ‘localized version’, ‘we would like’, ‘ would like to’,<br>‘like to offer’, ‘to offer software ’,‘offer software localized’, ‘software localized version’ |

Table 3.3: combination of variable length n-gram representation of an email

### 3.3.7 TF-IDF:

Raw text is inaccessible to machine learning algorithms. Rather, the text must be transformed into numerical vectors.

**Term Frequency (TF):** The total number of words in a document divided by the number of times a word appears in the document. Every document has its own frequency of words.

$$tf_{ij} = n_{ij} / \sum_{k} n_{ij} \tag{3.1}$$

**Inverse Data Frequency (IDF):** The log of the total number of documents divided by the total number of documents containing the word.w The weight of uncommon words is determined by inverse data frequency across all documents in the corpus.

$$idf(w) = log(N/dft) \tag{3.2}$$

Lastly, the TF-IDF is simply the TF multiplied by IDF.

$$w_{ij} = tf_{ij} * log(N/dft) \tag{3.3}$$

### 3.3.8 Word2vec:

The word2vec uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In CBOW we train to try to predict a word given a context. Here we take the whole corpus and we make a Bag of words model each will be a vector . Then the word vector is fed into the neural network of N neurons. The resultant N*VN*V matrix will then be passed through the softmax layer to get the probabilities. If we take the N*VN*V matrix the corresponding layer to that vector will be the vector for that word.

### 3.3.9 Training

Features that we obtained from the previous step were used to train the machine learning model by employing different popular classification algorithms. These algorithms are stochastic support vector machine(SVM), logistic regression (LR), decision tree (DT), random forest (RF), and multinomial naïve Bayes (MNB). We analyze these algorithms and explain their structure in our system in the following subsections:

#### 3.3.9.1 Support vector machine:

A support vector machine is a machine learning model that can generalize between two groups if the algorithm is given a set of labeled data in the training set. The SVM's primary purpose is to look for a hyper plane that can differentiate between the two groups. Many hyper planes can perform this job, but the aim is to find the hyper plane with the highest margin, which means the greatest distances.

$$w.x_i + b = 0 \tag{3.4}$$

where w is the weight factor and b is the bias. and x feature vector of sample i. SVM's work well on unstructured and semi-structured data, such as text, images, and trees, SVM's true strength is the kernel trick. We can solve any complex problem with the right kernel function.It handles high-dimensional data reasonably well.In reality, SVM models have generalization, and the probability of over-fitting is lower.SVM and ANN are often compared. SVMs outperform ANN models in terms of accuracy.

#### 3.3.9.2 logistic regression:

The binary classification problem lends itself well to logistic regression . The mathematical equation of the sigmoid function which has been provided below:

$$F(z) = 1/1 - e^{-}z \tag{3.5}$$

Now, in the above equation,

$$z = wo + w1.x1 + w2.x2 + ....... + wn.xn \tag{3.6}$$

As seen in the equation above, w0, w1 w2,..., wn, represents the regression of the model's co-efficient obtained by Maximum Likelihood Estimation, and x0, x1 x2,..., xn, represents the features or independent variables. Finally, F (z) in the above equation calculates the binary outcome likelihood, where the probabilities are divided into two categories based on the given data point (x).

#### 3.3.9.3 Decision Tree:

There are two kinds of nodes in the decision tree: external and internal. Internal nodes have the features required for classification, while external nodes reflect the decision class . The decision tree was analyzed using a top-down method, which divided homogeneous data into subsets. Its entropy, which is determined using Equation, defines the homogeneity of samples.The equation is given below:

$$E(s) = \sum_{l=1}^{n} p_i log_2 p_i \tag{3.7}$$

The probability of a sample in the training class is pi, and the entropy of the sample is E(S). The consistency of the split was determined using entropy. All of the features are taken into account during the split to determine the best split for each node. Permutation of the features is regulated by random state 0.

#### 3.3.9.4 Random Forest:

a random forest is made up of a large number of individual decision trees that work together as an ensemble. Each tree in the random forest produces a class prediction, and the class with the most votes becomes the prediction of our model.

**Random Forest pseudocode:**

1. Randomly select “k” features from total “m” features.
2. Where k « m
3. Among the “k” features, calculate the node “d” using the best split point.
4. Split the node into daughter nodes using the best split.
5. Repeat 1 to 3 steps until “l” number of nodes has been reached.
6. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

By comparing or combining the outcomes of various decision trees, it overcomes the issue of overfitting. Random forests perform better than a single decision tree for a wide variety of data items.The variance of a random forest is lower than that of a single decision tree. Random forests are very adaptable and have a high level of accuracy.

#### 3.3.9.5 Multinomial naive bayes:

The Multinomial Naive Bayes algorithm is a probabilistic learning method popular in Natural Language Processing (NLP). The algorithm predicts the tag of a file, such as an email or a newspaper post, using the Bayes theorem. It calculates each tag's probability for a given sample and outputs the tag with the highest probability. Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event. It is based on the following formula:

$$p(A|B) = P(A) * P(B|A)/P(A) \tag{3.8}$$

Where we are calculating the probability of class A when predictor B is already provided.

P(B) = prior probability of B

P(A) = prior probability of class A

P(B|A) = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text.The Multinomial Naive Bayes algorithm is worth studying because it has many applications in a variety of industries and makes real-time predictions. One of the most popular applications of the Naive Bayes algorithm is email classification.

### 3.3.10 Prediction

In this step, the trained classifier models have been used for classification.The test set has some mails which have been used to test the classifier. If the result is 1 then the mail is a SPAM email and if the result is 0 then the mail is a HAM email.

### 3.3.11 Implementation

The spam filtering system implementation consists of different modules.In section we will see the implementation details of this modules.

### 3.3.12 System Requirements

Our system takes email contents that means text as input and classifies it spam and ham.To implement this some hardware and software tools are needed. Required hardware and software tools are listed below:

### 3.3.13 Hardware requirements

From input to output the system propagate the following hardware:

- Minimum GPU RAM 8GB
- physical memory 32GB
- intel corei7-770k CPU
- Solid State Drive (SSD) 256GB
- Minimum 2h backup UPS
- Monitors

### 3.3.14 software requirements

We implement our system in a specific software environment.Required softwares are listed below;

- Operating system : windows 10
- python 3.6.9
- tensorflow gpu==2.2.1
- pandas==1.0.3
- Scikit.learn==0.22.2
- jupyter notebook

## 3.4 Conclusion

In this chapter, methodology for spam e-mail detection performing combination of variable length n-gram have been discussed.we have used two different feature extraction methods : n-gram and word2vec. where word-gram,character-gram and combination of variable length n-gram also performed and various classification methods have been used to classify emails.The next chapter is about the experimental result analysis of the proposed framework.

# Chapter 4

# Results and Discussions

## 4.1 Introduction

A thorough description of the proposed framework for spam email detection was provided in the previous chapter. The output of the proposed structure is examined in this chapter. With an Intel Core i5 processor and 8GB RAM, this framework was developed in a Python environment. The dataset 'spam or ham email' was used for this thesis.

## 4.2 Dataset Description

The 'spam or ham email' dataset is a collection of spam and ham emails.This dataset is collected from kaggle a online data publishing source which provides standard datasets. where there are 5731 emails.Of them 1369 emails are spam emails and the rest of the mails are ham emails.

The dataset mainly contains the contents of the emails.This is a labeled dataset where the spam emails are denote by'1' and ham emails are denoted by '0'.As this is a supervised machine learning problem, the dataset was divided into two portions, Training and Testing.We have used 80 percent mails for training and 20 percent mails for testing

## 4.3 Impact Analysis

Spam filtering has both social and ethical impact. As spammers getting smarter day by day they are adopting new techniques to send spam emails. Spam is a cause of e-mail-borne viruses, spyware, adware, and Trojan horses, as well as clogging Internet traffic by consuming a significant amount of network bandwidth. It's also used in denial-of-service, directory harvesting, and phishing attacks, both of which result in direct financial losses. Furthermore, spam also contains offensive, adult-oriented, and fraudulent materials that recipients find objectionable. Several anti-spam protocols are currently in use to differentiate spam from legitimate e-mails; nevertheless, spammers and phishers use complex spam structures to obfuscate email content and evade these procedures.A Spam filters prevent the servers from being overburdened with non-essential emails, as well as the more serious issue of being compromised with spam software, which could transform them into spam servers. So here lies the social and ethical impact of spam filtering.

### 4.3.1 Social Impact

Despite the fact that online social networks have become common communication platforms, survey findings show that email remains the most popular mode of Internet communication in our everyday lives. Spam emails are one of the most persistent issues in email systems. Spam emails, also known as spam emails, are unsolicited bulk emails that are sent to a large number of people. Spam emails normally hide the sender's address, and the spams are not requested by any of the recipients. Spam emails can include malware in the form of scripts or executable files, as well as hidden links to phishing websites.Again many confidential information can be leaked through spam emails. So here we can see that there is a great social impact of spam filtering in our day to day life.

### 4.3.2 Ethical Impact

Spam email is, at the very least, an annoyance that can clog your employees' inboxes and overburden your servers. Spam is also harmful because it serves as a gateway for serious attacks that can damage your machines, network, bottom line, and even your company's credibility. Yes, a spam filter solution is important as the first line of protection.So spam filtering plays a vital role in our life to protect our computer system as well as network system.In this way we can describe the ethical impact of spam filtering.

## 4.4 Evaluation of Framework

The primary objective of this project is to build a content based spam filter which could classify spam and ham emails. And also analyze the performance with various machine learning classifiers for features combination.To perform this experiment we have used open-source Google colab platform with Python == 3.6.9 and TensorFlow == 2.2.1 . Pandas == 1.0.3 data frame was used for dataset preparing and training and testing purpose, scikit-learn == 0.22.2 was also used. The training set is comprised of 80 percent of the total data and the testing set has 20 percent of the total data .During this section, we subsequently discuss the evaluation of performance and compare the result with other techniques. additionally, we compare the proposed model with existing techniques

**Measures of evaluation:** As various statistical and graphical measure are done to evaluate the system efficiency. So we have used the following terminologies for evaluation purpose:

- True positive(TP): Emails correctly classified as SPAM.
- False positive(FP):Emails incorrectly classified as SPAM.
- True Negative(TN):Emails correctly classified as as HAM.
- False Negative(FN): Emails incorrectly classified as HAM.
- Precision: It tells how many of the emails are actually spam among the emails that are classified as spam. Precision is calculated by Equation (4.1):

$$P = (TP)/(TP + FP) \tag{4.1}$$

- Recall: It gives the value of how many emails classified correctly as Spam among total spam emails. Recall can compute by using Equation (4.2):

$$R = (TP)/(TP + FN) \tag{4.2}$$

- f1-score: This is a useful evaluation metric to decide which classifier to choose among several classifiers. It is calculated by averaging precision and recall, which is done by Equation

$$R = 2 * P * R/P + R \tag{4.3}$$

The proposed framework's consistency is measured in the second step of implementation. The system creates a model based on the parameters and training data. The model is then fed data from the tests. For the samples of the testing data set, the model returns a predicted mark.

## 4.5 Evaluation of Performance

To evaluate the system performance we are using various machine learning methods like SVM,MNB,DT and LR.Again different feature extraction methods were applied to get better accuracy .

Table 4.1: Performance measure of the proposed framework for word-gram(bi-gram).

| bi-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.982 | 0.826 | 0.897 | 0.954 |
| Logistic regression | 0.971 | 0.728 | 0.832 | 0.929 |
| Decision tree | 0.790 | 0.942 | 0.859 | 0.925 |
| Naïve bayes | 1.0 | 0.742 | 0.852 | 0.938 |

Table 4.2: Performance measure of the proposed framework for word-gram(tri-gram).

| tri-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.967 | 0.746 | 0.842 | 0.932 |
| Logistic regression | 0.982 | 0.836 | 0.904 | 0.957 |
| Decision tree | 0.704 | 0.916 | 0.796 | 0.887 |
| Naïve bayes | 1.0 | 0.742 | 0.852 | 0.938 |

Table 4.3: Performance measure of the proposed framework for character-gram(bi-gram).

| bi-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.881 | 0.510 | 0.646 | 0.865 |
| Logistic regression | 0.861 | 0.474 | 0.612 | 0.855 |
| Decision tree | 0.714 | 0.572 | 0.635 | 0.842 |
| Naïve bayes | 1.0 | 0.742 | 0.852 | 0.938 |

Table 4.4: Performance measure of the proposed framework for character-gram(tri-gram).

| tri-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.975 | 0.855 | 0.911 | 0.959 |
| Logistic regression | 0.968 | 0.786 | 0.867 | 0.942 |
| Decision tree | 0.815 | 0.800 | 0.808 | 0.908 |
| Naïve bayes | 1.0 | 0.742 | 0.852 | 0.938 |

Table 4.5: Performance measure of the proposed framework for combination of variable length n-gram(1,2-gram).

| 1,2-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.988 | 0.913 | 0.949 | 0.976 |
| Logistic regression | 0.987 | 0.829 | 0.901 | 0.956 |
| Decision tree | 0.870 | 0.880 | 0.870 | 0.939 |
| Naïve bayes | 1.0 | 0.782 | 0.878 | 0.947 |

Table 4.6: Performance measure of the proposed framework for combination of variable length n-gram(1,4-gram).

| 1,4-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.984 | 0.891 | 0.935 | 0.970 |
| Logistic regression | 0.972 | 0.775 | 0.862 | 0.940 |
| Decision tree | 0.842 | 0.869 | 0.855 | 0.929 |
| Naïve bayes | 1.0 | 0.782 | 0.878 | 0.947 |

Table 4.7: Performance measure of the proposed framework for combination of variable length n-gram(2,5-gram).

| 2,5-gram | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.992 | 0.909 | 0.948 | 0.976 |
| Logistic regression | 0.987 | 0.851 | 0.914 | 0.961 |
| Decision tree | 0.879 | 0.873 | 0.876 | 0.940 |
| Naïve bayes | 0.565 | 0.993 | 0.720 | 0.894 |

Table 4.8: Performance measure of the proposed framework for word2vec .

| | precision | recall | F1 score | Accuracy |
|---|---|---|---|---|
| SVM | 0.913 | 0.076 | 0.140 | 0.776 |
| Logistic regression | 0.836 | 0.370 | 0.513 | 0.831 |
| Decision tree | 0.550 | 0.556 | 0.553 | 0.784 |
| Naïve bayes | 0.689 | 0.694 | 0.517 | 0.689 |

### 4.5.1 Confusion matrix:

A confusion matrix is a performance metric for machine learning classification problems with two or more classes as output.

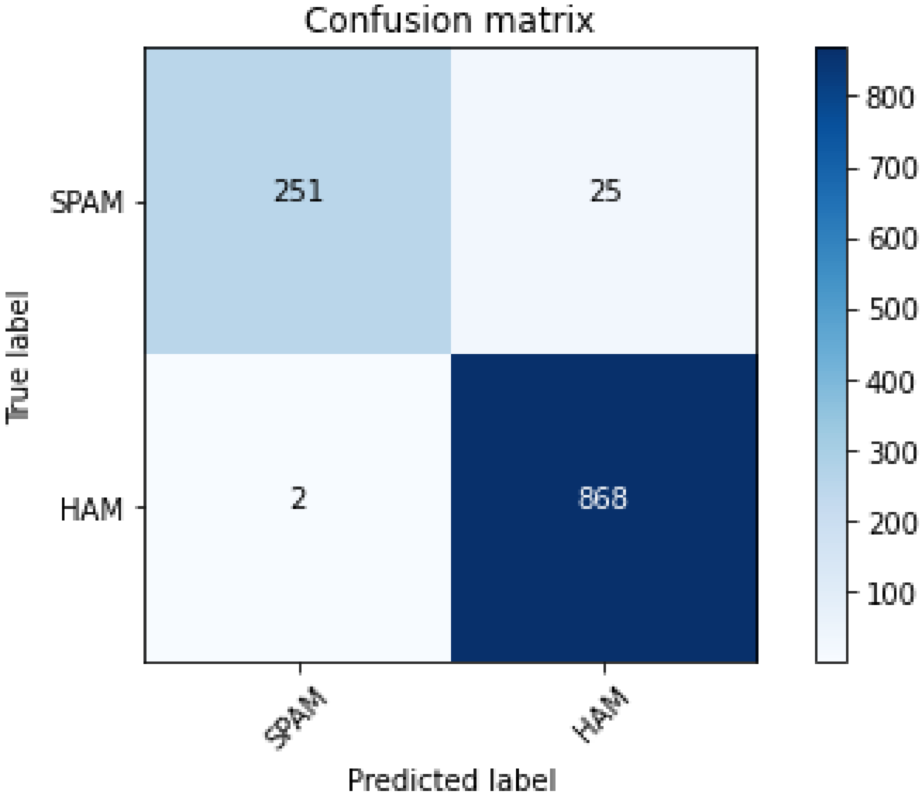


Figure 4.1: confusion matrix for proposed methodology

From the confusion matrix, we can say that

- True positive(TP): 868 emails are correctly classified as HAM.
- False positive(FP):25 emails are incorrectly classified as HAM.
- True Negative(TN):251 emails are correctly classified as SPAM.
- False Negative(FN):2 emails are incorrectly classified as SPAM.

### 4.5.2 ROC and precision-recall curve for the proposed methodology:

Precision-recall curve and ROC curve shows the graphical evaluation of algorithms. Precision recall curve shows value of precision and recall different threshold values.ROC curve shows relation between true positive rate and false positive rate.

### 4.5.3 ROC and Precision-recall curve for SVM classifier:

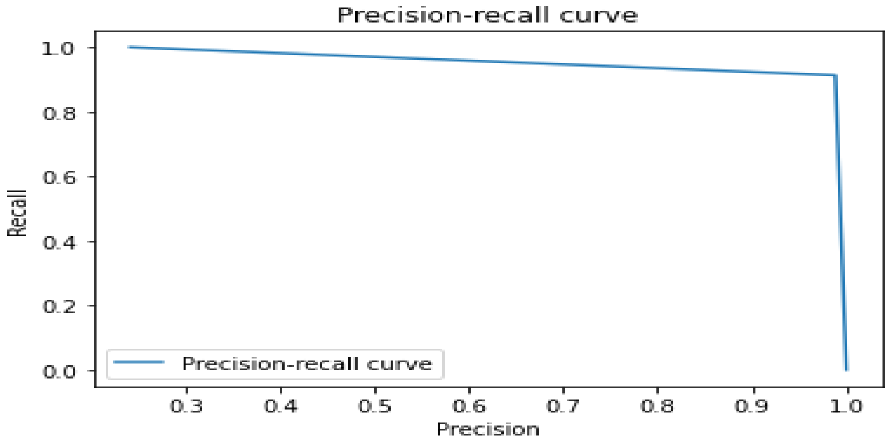


Figure 4.2: precision-recall curve for svc

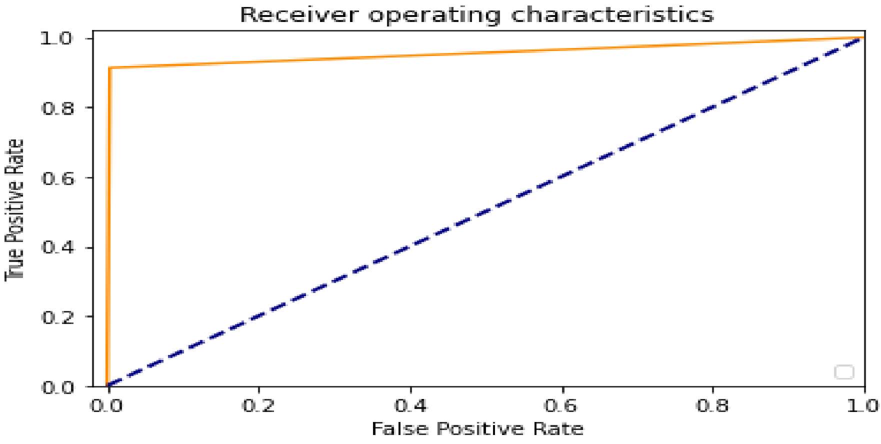


Figure 4.3: ROC curve for svc

### 4.5.4 ROC and Precision-recall curve for Logistic regression classifier:

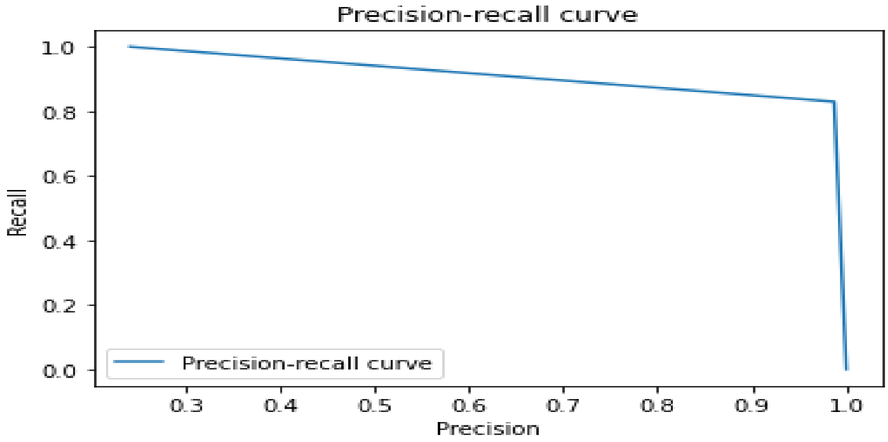


Figure 4.4: precision-recall curve for logistic regression

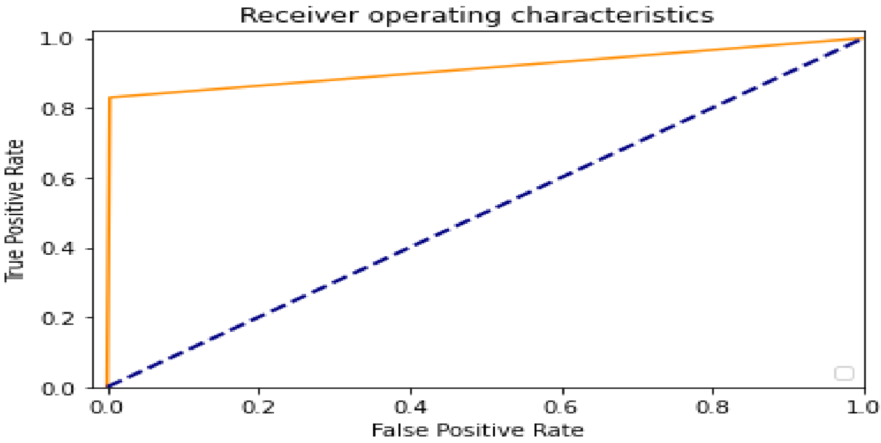


Figure 4.5: ROC curve for logistic regression

### 4.5.5 ROC and Precision-recall curve for Decision Tree classifier:

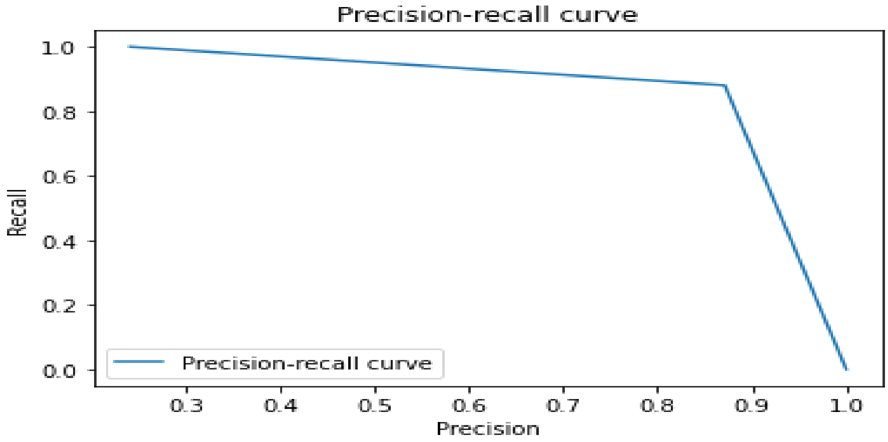


Figure 4.6: precision-recall curve for Decision Tree

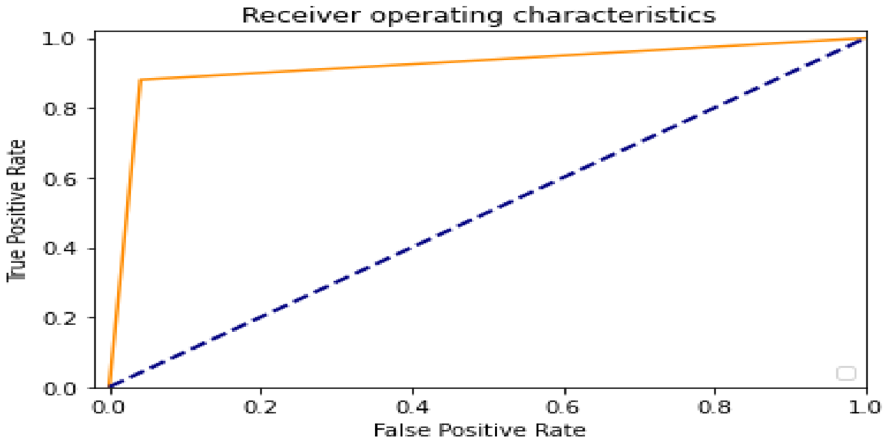


Figure 4.7: ROC curve for Decision Tree

### 4.5.6 ROC and Precision-recall curve for Naive Bayes classifier:

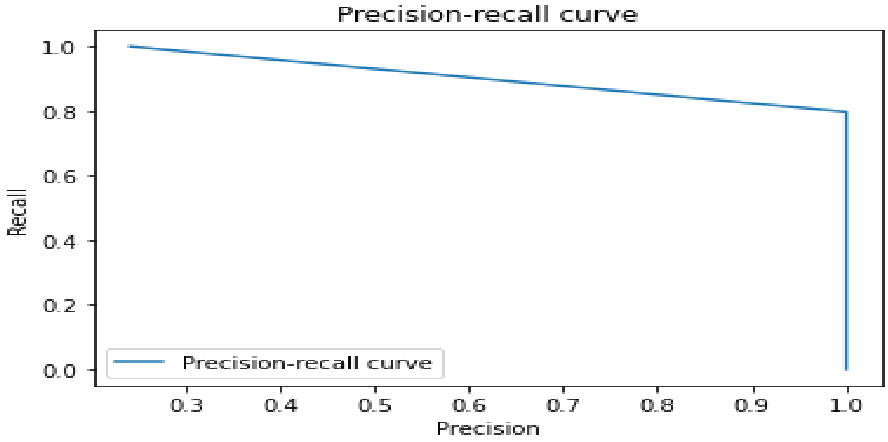


Figure 4.8: precision-recall curve for Naive bayes classifier

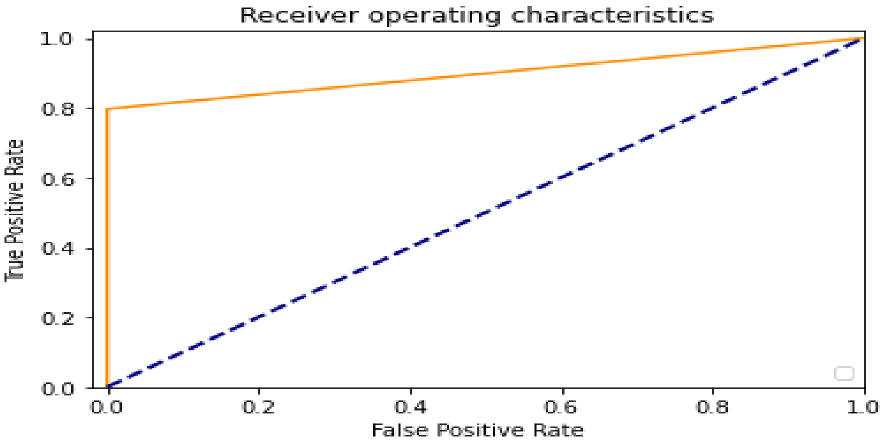


Figure 4.9: ROC curve for naive bayes classifier

### 4.5.7 Performance analysis for different feature extraction methods:

From the above analysis of various extraction methods we can see that combination variable length n-gram performs better than other methods.In[12] provided 80 percent accuracy with SVM classifier. Our proposed methodology provides accuracy of 97.6 percent with SVC classifier.

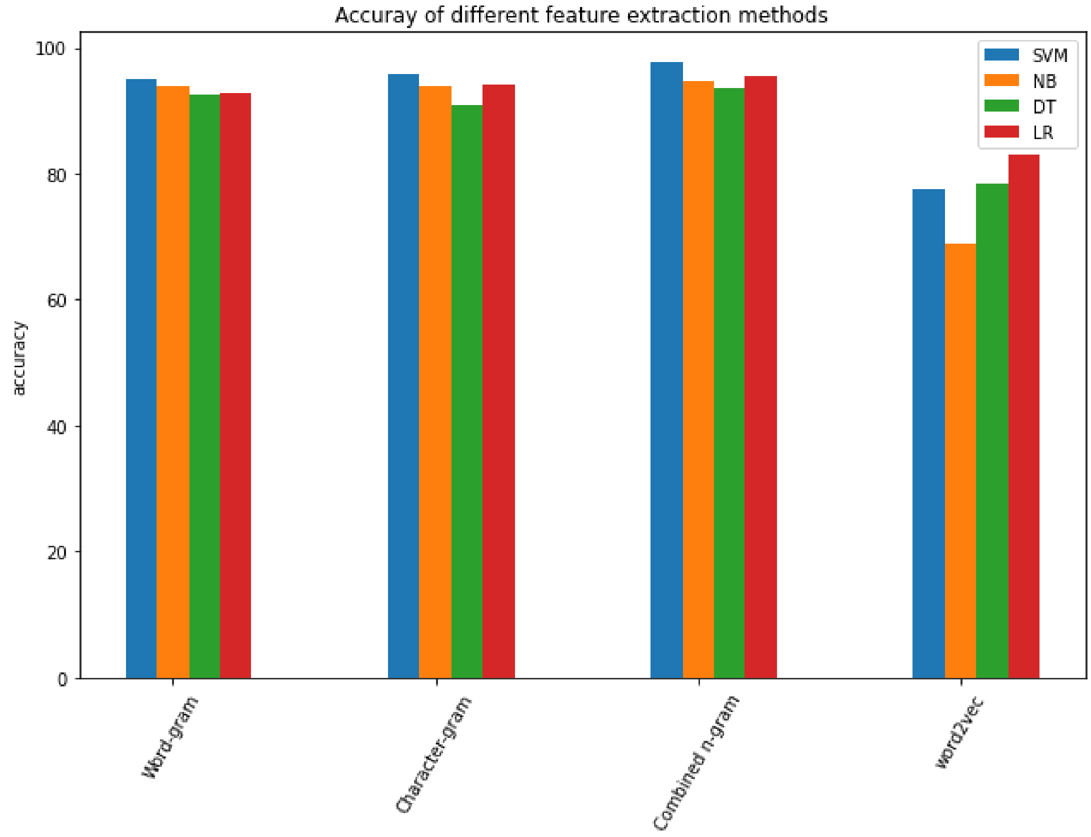
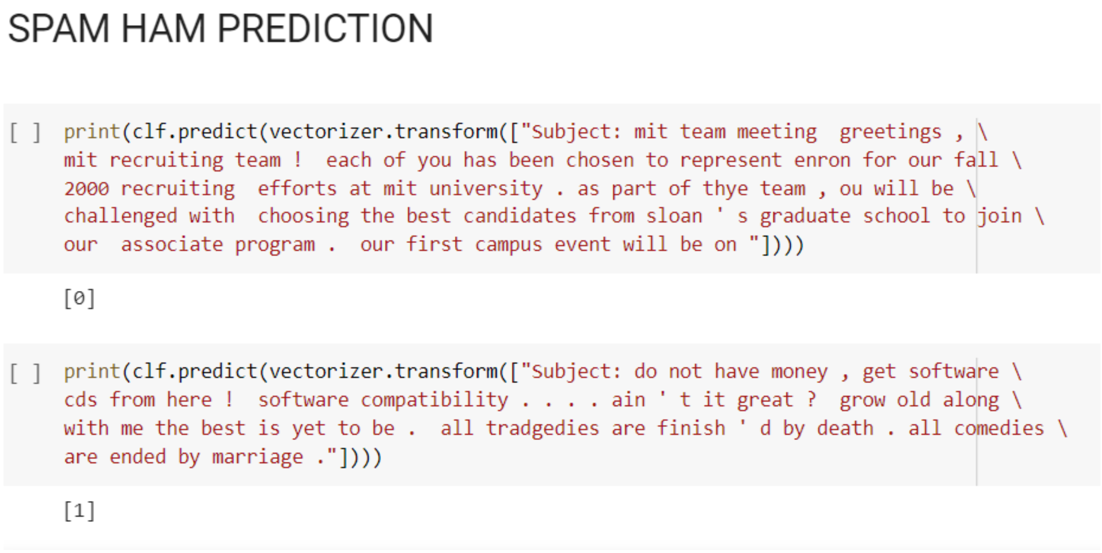


Figure 4.10: Comparative analysis of different methodology

### 4.5.8 sample input output:

In this section we provide the sample input output for the system:


SPAM HAM PREDICTION

```
[ ] print(clf.predict(vectorizer.transform(["Subject: mit team meeting  greetings , \
    mit recruiting team !  each of you has been chosen to represent enron for our fall \
    2000 recruiting  efforts at mit university . as part of thye team , ou will be \
    challenged with  choosing the best candidates from sloan ' s graduate school to join \
    our  associate program .  our first campus event will be on "])))

    [0]
```

```
[ ] print(clf.predict(vectorizer.transform(["Subject: do not have money , get software \
    cds from here !  software compatibility . . . . ain ' t it great ?  grow old along \
    with me the best is yet to be .  all tradgedies are finish ' d by death . all comedies \
    are ended by marriage ."])))

    [1]
```

Figure 4.11: sample input output of proposed methodology

## 4.6 Conclusion

The classification results for spam email detection are presented in this chapter. The proposed framework's performance is also addressed.The comparative analysis with different feature extraction techniques have also analyzed. The proposed combination of variable length ngram provides better accuracy as shown by the findings. The thesis work is brought to a close in the next chapter.

# Chapter 5

# Conclusion

## 5.1 Conclusion

Spam detection close to the sending server is a critical problem in network security, and machine learning techniques play a significant role in this field.Many solutions have been implemented to mitigate the negative effects of spam mail.Before we get into the project's main goal, it's important to understand some key aspects of spam messages. Spam is characterized as unsolicited email that contains ads or irrelevant content sent to multiple users who have not requested it, according to Information Technology. Spammers' main goal is to persuade consumers to purchase goods and services that are either legal or prohibitive, with the goal of making money.

In this project, our primary goal is to filter the messages in such a way that the users only read messages that are important to them based on the email body using different feature extraction technique. The extracted features are analyzed using machine learning classification. We explore three different supervised machine learning algorithms, namely, naïve-Bayes, logistic regression , support vector machine and Decision Tree.

In our proposed approach, the n-grams are generated from the training dataset and are preserved accordingly to predict the unlabeled data for the classifiers.However, our contribution is to perform combination of variable length n-gram.We have also performed word-gram, character-gram and word2vec method for comparative analysis.Our models provides 97.6 percent of accuracy with SVM classifier.which provides better accuracy than others.

## 5.2 Future Work

It is impossible for a spam filtering solution to be 100 percent accurate. As a result, our spam filter isn't particularly weak, but it can certainly be improved by integrating different features into the filter. As a result, there are scope for improvement in the future. The following are some potential enhancements:

- Spammers are now luring users with image-based spam. To detect image-based spam, image-based spam detection software can be created.
- Spam is now also being used in the Bangla language. It is possible to detect spam emails in the Bangla language using language processing techniques.
- The message in a modern email system includes HTML info. HTML tags were not taken into account. HTML tags and comments can also be used as tokens to detect intelligent spam.
- Furthermore, the majority of anti-spam email filtering methods currently in use are client-side based.As a result, once the email has been sent to the recipient, all of the emails are confidential. The sending and receiving method already degrades the performance of networks and servers. As a result, if email can be classified before it is sent to a recipient, it can help to minimize network and server workload.

# References

[1] P. Liu and T. Moh, 'Content based spam e-mail filtering,' in *2016 International Conference on Collaboration Technologies and Systems (CTS)*, 2016, pp. 218–224. DOI: `10.1109/CTS.2016.0052` (cit. on p. 6).

[2] D. Gaurav, S. M. Tiwari, A. Goyal, N. Gandhi and A. Abraham, 'Machine intelligence-based algorithms for spam filtering on document labeling,' *Soft Computing*, vol. 24, no. 13, pp. 9625–9638, 2020 (cit. on p. 6).

[3] I. Kanaris, K. Kanaris and E. Stamatatos, 'Spam detection using character n-grams,' in *Hellenic conference on artificial intelligence*, Springer, 2006, pp. 95–104 (cit. on p. 7).

[4] C. C. Kiliroor and C. Valliyammai, 'Social context based naive bayes filtering of spam messages from online social networks,' in *Soft computing in data analytics*, Springer, 2019, pp. 699–706 (cit. on p. 7).

[5] W. Feng, J. Sun, L. Zhang, C. Cao and Q. Yang, 'A support vector machine based naive bayes algorithm for spam filtering,' in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, IEEE, 2016, pp. 1–8 (cit. on p. 7).

[6] M. Rathi and V. Pareek, 'Spam mail detection through data mining-a comparative performance analysis,' *International Journal of Modern Education and Computer Science*, vol. 5, no. 12, p. 31, 2013 (cit. on p. 7).

[7] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz and F. C. Garcıa, 'Content based sms spam filtering,' in *Proceedings of the 2006 ACM symposium on Document engineering*, 2006, pp. 107–114 (cit. on p. 7).

[8] J. Moon, T. Shon, J. Seo, J. Kim and J. Seo, 'An approach for spam e-mail detection with support vector machine and n-gram indexing,' in *International Symposium on Computer and Information Sciences*, Springer, 2004, pp. 351–362 (cit. on p. 7).

[9] S. Kaur, 'Spam detection using n-gram analysis and machine learning techniques,' 2019 (cit. on p. 8).

[10] N. F. Rusland, N. Wahid, S. Kasim and H. Hafit, 'Analysis of naıve bayes algorithm for email spam filtering across multiple datasets,' in *IOP conference series: materials science and engineering*, IOP Publishing, vol. 226, 2017, p. 012 091 (cit. on p. 8).

[11] D. S. Jawale, A. G. Mahajan, K. R. Shinkar and V. V. Katdare, 'Hybrid spam detection using machine learning,' *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 2, pp. 2828–2832, 2018 (cit. on p. 8).

[12] H. Bhuiyan, A. Ashiquzzaman, T. I. Juthi, S. Biswas and J. Ara, ‘A survey of existing e-mail spam filtering methods considering machine learning techniques,’ *Global Journal of Computer Science and Technology*, 2018 (cit. on p. 8).

[13] T. M. Ma, K. YAMAMORI and A. Thida, ‘A comparative approach to naıve bayes classifier and support vector machine for email spam classification,’ in *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, IEEE, 2020, pp. 324–326 (cit. on p. 8).

[14] B. K. Dedeturk and B. Akay, ‘Spam filtering using a logistic regression model trained by an artificial bee colony algorithm,’ *Applied Soft Computing*, vol. 91, p. 106 229, 2020 (cit. on p. 9).

[15] L. Bansal and N. Tiwari, ‘Feature selection based classification of spams using fuzzy support vector machine,’ in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, 2020, pp. 258–263 (cit. on p. 9).

[16] S. Abiramasundari, V. Ramaswamy and J. Sangeetha, ‘Spam filtering using semantic and rule based model via supervised learning,’ *Annals of the Romanian Society for Cell Biology*, pp. 3975–3992, 2021 (cit. on p. 9).

[17] S. B. S. Ahmad, M. Rafie and S. M. Ghorabie, ‘Spam detection on twitter using a support vector machine and users’ features by identifying their interactions,’ *Multimedia Tools and Applications*, pp. 1–23, 2021 (cit. on p. 9).

[18] R. Nayak, S. A. Jiwani and B. Rajitha, ‘Spam email detection using machine learning algorithm,’ *Materials Today: Proceedings*, 2021 (cit. on p. 9).

[19] J.-J. Sheu, K.-T. Chu, N.-F. Li and C.-C. Lee, ‘An efficient incremental learning mechanism for tracking concept drift in spam filtering,’ *PloS one*, vol. 12, no. 2, e0171518, 2017 (cit. on p. 10).

[20] A. Barushka and P. Hajek, ‘Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks,’ *Applied Intelligence*, vol. 48, no. 10, pp. 3538–3556, 2018 (cit. on p. 10).

[21] J.-J. Sheu, Y.-K. Chen, K.-T. Chu, J.-H. Tang and W.-P. Yang, ‘An intelligent three-phase spam filtering method based on decision tree data mining,’ *Security and Communication Networks*, vol. 9, no. 17, pp. 4013–4026, 2016 (cit. on p. 10).

[22] M. Hassan and M. W. Hussain, ‘Header based spam filtering using machine learning approach,’ *International Journal of Emerging Technologies in Engineering Research*, vol. 5, no. 10, pp. 133–140, 2017 (cit. on p. 10).

[23] J. R. Mendez, T. R. Cotos-Yanez and D. Ruano-Ordas, ‘A new semantic-based feature selection method for spam filtering,’ *Applied Soft Computing*, vol. 76, pp. 89–104, 2019 (cit. on p. 10).

[24] S. Kumar, X. Gao, I. Welch and M. Mansoori, ‘A machine learning based web spam filtering approach,’ in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, 2016, pp. 973–980 (cit. on p. 10).

[25] S. Nagaroor and G. Patil, ‘Mitigating spam emails menace using hybrid spam filtering approach,’ in *International Conference on Emerging Research in Computing, Information, Communication and Applications*, Springer, 2016, pp. 219–227 (cit. on p. 11).