IMPROVING THE DETECTION OF EMAIL SPAM FILTER USING LGS-COUNT MODEL

MSc Research Project

MSc in Cybersecurity

Shiva Prasad Bonu

Student ID: 20169850

School of Computing

National College of Ireland

Supervisor: Imran khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: BONU SHIVA PRASAD

Student ID: X20169850

Programme: MSc in Cybersecurity **Year:** 2020-2021

Module: RESEARCH PROJECT

Lecturer: IMRAN KHAN

Submission Due Date: ……31-01-2022……..……

Project Title: IMPROVING THE DETECTION OF EMAIL SPAM FILTER USING LGS-COUNT MODEL

Word Count: **7886** **Page Count:** ……28…….……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……BONU SHIVA PRASAD……

Date: ……31-01-2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

The Differences:

The main difference with the existing models the project is which is better code and accuracy is more. As we see we have compared even with other models.

Novelty Aspects of the project

Based on the project the research done the algorithm Used from different authors making best of it. The massive usage of the web application where the TG-IDF algorithm with high accuracy. Which helps to filter all spam filters with even helps to fall in wrong hands. Compared to all methods done which method which gave high accuracy even when the ratios are changed. We used Adaboost and random forest and logical regression. The data proves us logical regression is highly efficient. The ML algorithm which helps to filter the data and give high accuracy.

IMPROVING THE DETECTION OF EMAIL SPAM FILTER USING LGS-COUNT MODEL

Bonu Shiva Prasad

X20169850

Abstract

In today's advent with an increase in web popularity, there has been an increase in its usage and data amongst end users. In such a scenario, e-mails have become one of the most secure medium to make online transactions to fulfil the purpose of communication and transfer required data. Due to its convenient nature of use this had led to a significant revolution taking place over conventional communication systems. However, the main obstruction behind mails is the publication of unwanted and harmful mails known as spam. Spam mails are deceptive mails that are intentionally sent to cause harm to the end user. Hence a detection method to avoid such scenarios is needed. Spam mails are generally detected through ML and NLP mechanisms and therefore this thesis puts forward the working principle of TF-IDF and stemming algorithms to detect such words and further classify it as spam mails (unwanted) and ham mails (valid). The working implementation of the thesis is carried out on the CSDMC 2010 dataset. Further, the training and testing process is executed and the proposed method is implemented. The thesis focuses to develop an enhanced spam exposure framework based on count vectorizer and TF-DF vectorizer. Lastly, the classification of spam and ham mails are evaluated using a comprehensive range of ML algorithms and results are calculated based on ROC curves and confusion matrix.

1 Introduction

In recent years, the internet platform has provided multiple services to its end users; making it a necessity in day-to-day life. One such service is emails. It's considered to be one of the substantial platforms on the web used for communication purpose and transfer of data and information. The acronym of email stands for electronic messaging framework; that forms as the typical and the most conventional methods used for the purpose of information communication. It has many branches such as Google and Outlook. However, the normal implementation of this gets obstructed sometimes and results into the presence of hazardous and malicious files in the users system. The destructive nature of such mails results into the generation of spams. Such mails are believed to be junk mails that are send by the spammer with an intention to cause damage to an individual's networking system. Spam mails also result into increased storage space and decreased computational power. Hence detecting such mails and filtering them is essential to create an effective working environment.

The process of spam filtering generally involves detecting unsolicited messages and preventing them from occupying space in the user's mailbox. On the other hand, the developers have also been creating multiple anti-spam detection techniques to prevent the existence of bulk mail.

1.1 Background

No doubt, internet has become an essential part of our lives and email services provided by the web platform has proven to be an effective tool for communication purpose. However, an increased observation on spam mails has also been equally on rise and can mitigate from any part of the industry. Primarily, every industry and organisation faces the issue of spam mails as they are connected to the web. Hence, it becomes the utmost need to address the issue and generate a solution. For this, the developers must look into the system and further prevent it from occurrence of such mails. Portal filtering is one such method, in which spams are encouraged to be filtered out and not reach the end users. Unsolicited Bank Email (UBE) also known as spams comprises of redundant messages sent by a spammer to an organisation on a regular basis. Keeping this scenario in mind, spam filtering becomes a necessary task. But however, a disadvantage of applying spam filtering technique is that, in the process of discarding spam mails, valid mails are also deleted. Hence, this negatively affects the working procedure of spam filtering.

Hence, it is advised that spam filters be applied to all the layers, as firewalls might be present on the mail servers providing mail security before a spam is received through a potential network. These filters can also be executed on customers end by automatically discarding irrelevant messages based on the mentioned criteria of the message. Below are some problems faced in spam filters:

- **Cost** – the presence of spams on a computer result in loss of network data and increased storage bandwidth. Hence, eliminating spam mails globally might involve large amounts of monetary factors to deal with the issue globally.
- **Privacy** – the working of spammers involve, sending links of fake websites which can be accessed only through user credentials. Hence forcing the user to provide his personal details, this can be misused by the spammer.
- **Security** – as the spam mails occupy a larger space in the users account, the security of the server is put to risk. Since all the private information of the user is now accessible by the spammer, the overall security if the system is compromised upon.

1.2 Problem Statement

With technology becoming a vital part of our lives, this century has witnessed the usage of internet reaching to its exponential heights. One such commonly used web usage is sharing of mails to transfer data and information amongst individuals. While such mails are necessary for everyone, they come along with unnecessary bulk mails known as spams. These spams are responsible to occupy majority of the systems storage and increase the bandwidth of the server, thereby decreasing its computational power. Also such mails, divert the attention from legit mails to fake mails and directs an individual towards detrimental solutions. One of the major problems is that spam mails reduces the overall speed of the net and decreases the optimization power of the system. These mails also have the potential to corrupt the working system of an organisation by smuggling potential viruses into the system. Also, identification of such mails becomes a tedious task. On one hand, where detection of such spams can take

place manually, there spam filtering involves large amount of time to be invested in it. Hence, detecting the presence of such spams becomes the need of the hour.

1.3 Motivation

An inclined rise has been witnessed of spam mails since the 1990's. Apart from the problems mentioned above, these mails comprises of fake links that leads a user to phishing sites. These sites generally comprise of malwares and are triggered on entering user's sensitive information. Also, the processing of such unstructured mails has become a tedious task in organisations. Therefore, the detection of such spams has been encouraged using stemming methodologies. This has motivated us to utilize the concepts of TF-IDF and CV along with ML strategies, and extend spam filtering and detection techniques that could further classify mails as spam or ham. The proposed thesis is implemented using four machine learning algorithms namely: Naiive Bayes, logistic regression, AdaBoost and random forest.

1.4 Research Questions

To accomplish the purpose of the project, below are the guided research questions:

- On what factors does the occurrence of spam takes place on the server system?
- What are the existing methods to detect spam mails?
- Do the existing methods correctly classify spam and ham mails?
- What amount of dataset should be fed to the training and testing phase?
- Will the concept of TF-IDF work best to detect spam mails?
- Which machine learning algorithms can be effectively used?
- Will the comparison between ML algorithms carried out on the concepts of CV and TF-IDF produce results as expected?
- What shall be the maximum accuracy that can be accomplished?

1.5 Organization of Thesis

The aim of the thesis is to enlighten network concerns that may arise due to existence of spam mails in organisation servers. The introductory part of the thesis summarizes the existing issues of spam mails and puts forward the research questions that shall navigate throughout the implementation of the work. Chapter 2 puts forward the literature survey carried out by multiple authors in the field of detecting spam mails. Chapter 3 discusses the methodologies used to implement the work along with the concepts of machine learning and TF-IDF. Chapter 4 briefs the design specifications along with the workflow of the model. Chapter 5 gives a detailed summary on implementation and associated terminology with it, followed by results and evaluation metrics in chapter 6. The thesis is brought to end with conclusions taking place in chapter 7 and finally concluded with references.

2 Related Works

This section summarizes similar work performed by multiple authors in the field of detecting spam mails using machine learning algorithms.

(Govil.et.al) proposed his theory of detecting spam mails through filters. These filters could further predict and classify spam and non-spam mails. The implementation mechanism was carried out after researching the nature of emails. Since the mails are received on the net, it has a vast exposure to spammers and phishers. Mails send by the spammers are very sensitive in nature and contain links that might lead to phishing sites. Hence, a prudent mechanism was proposed by him that could detect the existence of such mails.

(Amani Alzahrani.et.al) put forward his research work that focused on spam mails that were being received via SMS on mobile devices. These SMS messages had their origins on advertising companies who exploited the popularity of emails by spreading unwanted mails containing their ads. These advertisement mails also included offers, near discount values and extra availing services. However, the quantity in which the companies sent these messages seemed to often frustrate the user. As these mails occupied majority of the memory space and decreased their server computational power. Hence, developers started taking measure and used the fundamentals of machine learning and neural networks to detect spam mails and classify them as spam or ham. The most widely used ML methods included Naiive Bayes and logistic regression.

(Hezha.et.al) focused on the importance of emails and how they provided ways to transfer data and information securely. These emails were popularly known for their secure and reliable mode of communication transfer. However, the author also observed that these mails resulted into generation of unwanted and irrelevant data on the mail holder's account. This unwanted and junk data was labelled as spams. These mails created similar copies of unnecessary data sent from anonymous users and phishers. Once these spam mails are received on the users end, it occupies a majority of their memory space and reduces the computational speed of the server. Although the developers created filters that could address this issue and thereby reduce the risk of malwares so being created. Through these methods developers tried to increase the awareness amongst end users and generated a preventive mechanism that could further protect an individual's system. The authors in this work proposed string matching algorithms to detect the existence of such mails. Further the authors examined and compared six string matching algorithms. The algorithms included the concepts of TF-IDF and longest common subsequence (LGS).

At times the implementations of spams are intentionally done to demoralize the popularity of an organisation. Such spammers are even paid to do so. They are given monetary incentives to make an exception and defame a company. Although this process is not considered ethical, but however it pays off spammers and continues to exist as a certain business.

(Karthika Renuka.et.al) understood the importance of mails and how it provided worldwide accessibility to communication and information transfer. However, failures of implementing standard mail protocols, an increase in e-business and risks involved in financial transactions have contributed to increase mail threats. The authors conducted a survey, so as to how this threat might result in spam; as it may cause financial damage to organisations and customers. Such spam mails invades the customers privacy and gets added into the mail containers

without the consent of the user. This results into a decrease in speed and increased computational capacity of networks being utilized on the users end. Despite the users openly rejecting spam mails, it continues to exist in every mail box and serves as a source of income for many spammers.

(Ghulam Mujtaba.et.al) presented a study wherein they focused on how mail services had their control in major parts of the organisation that required information communication. Hence, this resulted the mails to form a crucial part of every transaction that took place between business customers. Despite the existence and proliferation of alternative ways to emails, its relevance was never lowered; instead an increase in messaging exchange was observed. Hence, as this volume of exchange of mails increased, so did the need to automate email management increased. The reason of increased mail management was done with the purpose of detecting spams and phishing attacks.

3 Research Methodologies

In an advent of massive usage of net applications, the worldwide use of mails has been increasing widely. And so does the associated risks involved in it. Hence, the classification of such mails into spam or nor spam has become a global issue as it not only requires computerized amount of time, but also large manpower. As a result of which many developers began generating mail detection techniques that could classify and segregate the junk mails with the legit ones *(Ni Zhang.et.al)*.

In this thesis, a machine learning based approach has been put forward that takes full advantage of a larger dataset and act as scalable alternatives to the existing ones. The thesis focuses on four ML algorithms namely: Naiive Bayes, logistic regression, Adaboost and random forest. These algorithms are further accompanied with concepts of TF-IDF and Count vectorizers. This classification, using ML algorithms has provided mail management techniques that could handle large production of spam mails with greater accuracy. Hence, many researchers have worked on this concept of spam detection that is used to solve complex issues; that could in return satisfy user experience *(Youn.et.al)*.

However, the summary of the proposed approach is to enhance the working implementation of spam detection using ML algorithms and extend this range of execution so as to accomplish higher accuracy. This section of the thesis focuses on the methodologies so adapted to implement the same.

3.1 Working

The implementation of the thesis takes place in two stages: classification and action. The classification stage is responsible to classify mails into spam or ham and detect whether the messages are harmful or not. The second stage of implementation that focuses on action is responsible to further reject the classified mail or mark it forward for transfer. The working of the entire thesis can be summarised in the below steps:

Step1: collect spam and not spam data from repository

Step2: pre-process the dataset and remove further redundant data from it

Step3: check the sender of the mail/source of origin

Step4: apply required classification strategies according to the requirement of the application

Step5: evaluate on the results stored

The below diagram illustrates the anti-spam flowchart:

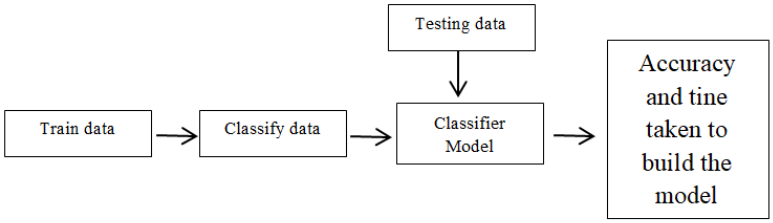

Figure1: Anti-Spam Flowchart

3.2 Description

In order to successfully detect a spam, it is important to comprehend, what is the type of spam. To select the type of spam, the first action needed is to choose a file from dataset repository and perform feature extraction techniques on it. To accomplish the feature extraction effectively “count the sentence” approach can be implemented. The next step involves, inspecting information of the dataset using ML classifiers that could further classify and detect whether the mails are spam or ham. The outcomes of this result are further used to predict the final status of the mails. Diagram below depicts the workflow of the proposed method.

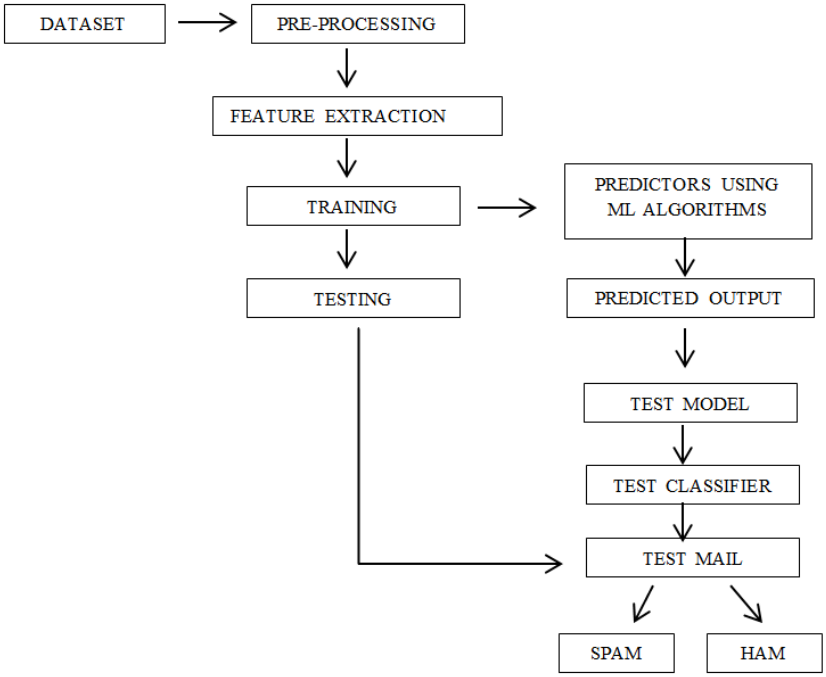

Figure2: Workflow of the proposed method

3.3 Pre-Processing

An email is generally represented as a collection of feature vectors. These vectors together represent huge number of files that are used to develop a vector matrix. These matrix vectors

are sparse in nature and are large in quantity due to its mail files. This vector matrix further undergoes dimensionality reduction followed by classification of spam mails *(Zhang.et.al)*. The pre-processing technique involves the removal of noise and redundant data from the dataset which might not be required and used further. The technique of pre-processing generally includes:

- Elimination of irrelevant data such as numbers and symbols
- Elimination of unnecessary URLS's
- Elimination of redundant words using word stemming

The diagram below represents a typical pre-processing phase.

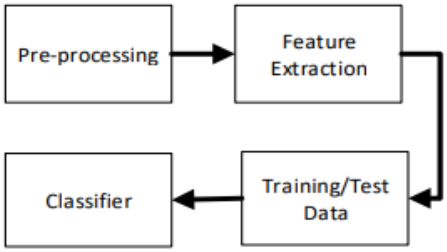

Figure 3: Steps involved in Pre-Processing

3.4 Feature Extraction

The process of feature extraction includes extracting relevant features from the dataset and further transforming these features into a 2D vector representation of functional space; wherein all the features are mapped accordingly. Further, this feature vectors are defined using the concepts of TFI-DF values *(Laorden.et.al)*. Later, this process also involves cleaning and tokenization phase wherein; a normal set of words are converted to set of feature vectors that are understandable by the ML algorithms. These ML algorithms are further responsible to classify mails as spams or not spam.

3.5 Test Classifier

Once the data is fed into the predictors, it output from this is further given for the testing phase. This process is necessary to carry out, as the accuracy of the classifier is tested using certain parameters and finally evaluated against a set of evaluation metrics *(Sanz.et.al)*.

3.6 Test Mail

Once the process of training is completed, the new mails are then fed as input to the classifier. This classifier produces the final output as 0's and 1's wherein; the 0 represents ham mails and 1 represents spam mails.

3.7 Classification

The primary purpose is achieved using classification techniques that filter out the messages to accomplish higher efficiency. In the proposed method four machine learning algorithms are put forward to classify the model and predict mails.

4 Design Specifications

The primary aim of the thesis is to carry out a spam detection technique and classify it as junk or legit using ML algorithms. For this purpose, the information source is fed to test classifiers that were initially obtained from the dataset. This dataset contains spam mails and ham mails. The diagram below represents the architectural framework of the proposed method.

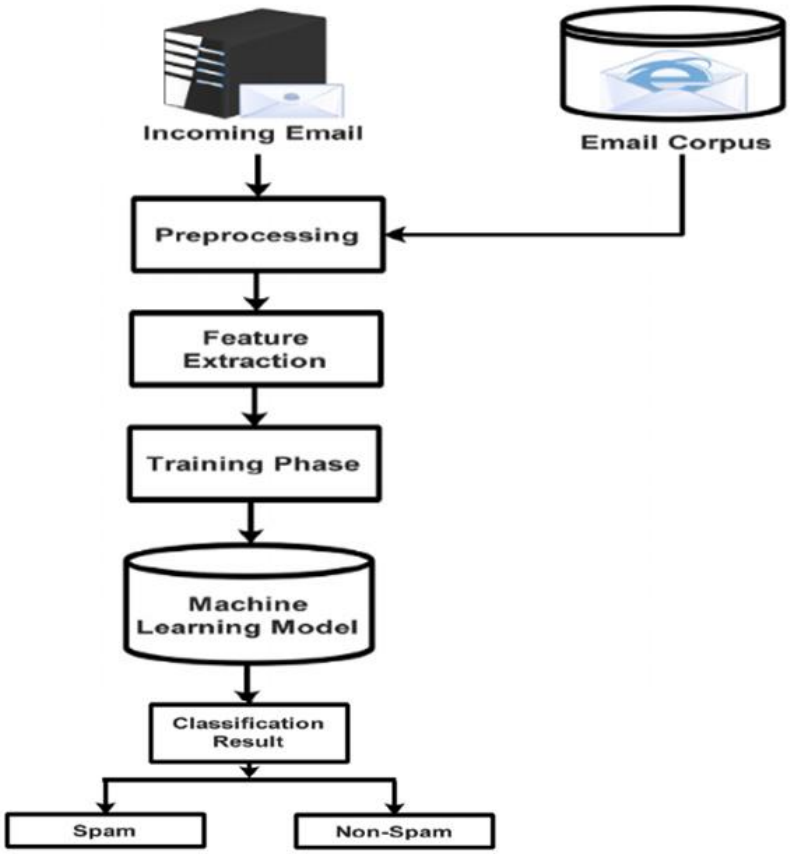

Figure 4: Architectural framework of the proposed method

4.1 Configurations

Below are the configurations used to implement the thesis:

- The machine used for development = Dell XPS Intel i7 processor
- RAM = 16GB
- OS: Windows 10
- Language = Python 6.3.3
- Text Editor = VSCode

4.2 Libraries

- Pandas
- Numpy
- Sklearn
- Matplotlib

- wxPython

4.3 Text Processing

This is the initial step of entire processing to accomplish the goal of the thesis. It's generally a short procedure in which the structure of the email and its content is extracted and further converted to plain text for text analysis.

4.4 Feature Set and Vectorization

The implementation of the proposed thesis puts forward two feature sets:

- Words using TFI-DF frequency
- Words using Count Vectorization

Both the above feature sets are developed and generated using the same kernel as that involved in the text processing phase.

4.5 Tokenization

The process of tokenization involves breaking a stream of words into meaningful phrases and symbols that are called as tokens. This process generally occurs at word level and possess the below characteristics:

- Continuous string of alphabetic characters are considered to be one token
- Tokens are usually separated by whitespace characters
- Punctuations and white spaces are also included as tokens

4.6 Stemming

Stemming is a process used to decrease a word to its root word or stem word that is known as Lemma. When a new word is found, it offers opportunities for further research work and extensions. Stemming is generally performed by individual algorithms and further belongs to the AI branch of networks. However, simple algorithms are generally used to strip of original words by recognizing its prefixes and suffixes in order to reach to the root word.

4.7 Naiive Bayes Spam Classification

The concept of Naiive Bayes model is that, it predicts all of the entities of its model are independent of each other, hence the word "naiive". It can be calculated using conditional probability given as:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

The assumption that an NB model uses is that the incoming words are completely independent to each other. This results, that each word in the dataset can be segregated as spam or ham, and is calculated as the likelihood of the product. The multinomial NB forms as one of the three types of NB algorithms that make use of discrete counts to detect the presence of spam mails.

4.8 Random Forest

A random forest is a collection of decision tree whose patterns are very much similar to that of a tree structure. It is considered to be a distinctive technique that leads to gain knowledge on every classification being performed. Each node of a random forest represents a leaf node that has an intended feature value, and every feature value of a certain set of trees represents a branch of a sub tree. This method is generally used to derive classification related issues. The beginning of every tree is marked by a root node.

4.9 AdaBoost

In places where boosting combines weak learners with those of strong learners, gradient boosting creates an ML model by merging all the weak learners and training them repeatedly until a point of minimum loss is reached. The steps to execute this concept are as follows:

- Surveying the data for errors
- Remove the errors that induce overfitting issues
- Further, improve the classification model to increase the accuracy
- Finally, assign equal weights to all the classifiers

5 Implementation Details

5.1 Dataset

The proposed dataset has been taken from CSDMC2010 SPAM corpus, through Kaggle repository. This dataset contains spam and ham data folders with a spam count of 2332 files and ham count of 1083 mail files. Rather than making use of complicated hybrid models, our proposed approach utilizes simple working ML algorithms along with the concept of TF-IDF and CV. The dataset which is in the HTML format is further converted into plain text format using the text-processing technique. However, the paper makes use of two feature sets:

- TFI-DF
- Count Vectorization

Below is the count model of dataset containing spam and ham files:

Figure 5: Dataset of spam and ham files

5.2 Integration of TFI-DF and CV

The primary aim of the proposed thesis is to create a method that could detect the presence of spam mail files in the mail box of a user. The implementation of the thesis is carried out using the concepts of TFI-DF and CV that are further accompanied with machine learning classifiers. The entire processing takes place on the dataset so acquired from Kaggle

repository containing large number of spam and ham files. Later, fundamentals of stemming, tokenization and text –processing are also applied to filter irrelevant words. Finally the overall implementation is evaluated on the basis of performance metrics such as ROC curves and precision-recall factor. The diagram below depicts the perfect implementation of the thesis and illustrates how the model works by incorporating all ML classifiers using TFI-DF and CV concepts.

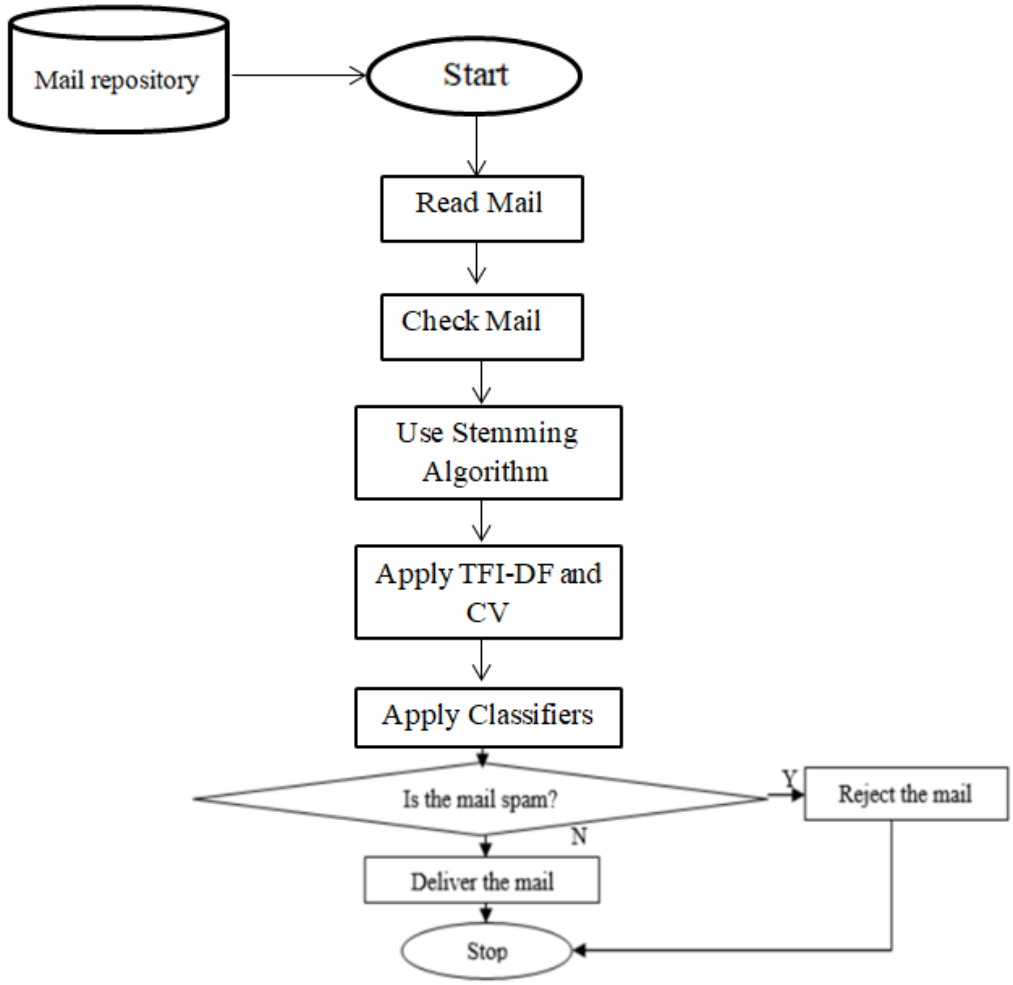

Figure 6: Implementation flow of mail filtration

The main idea behind incorporating TFI-DF and CV with ML classifiers using stem algorithms is to detect the presence of spam mails and further classify them to increase the overall speed of the processing systems and achieve higher accuracy. As the primary goal of the thesis is to find a method that could detect valid terms via stemming algorithm. This algorithm works by eliminating stop words and detecting valid words that are further stemmed to their roots using stemming algorithm. The next step involves calculating term frequencies and relevance of valid terms. Next these terms are used to form a vector table using TFI-DF methods. This formation of vector table takes place in pre-processing phase and hence reduces the processing time. Finally, implementation of TFI-DF takes place. As depicted in figure above, the process starts by receiving mail in user's mail box. Next the mail is checked for repetitive words that would not contribute to further prediction of spam mails. After this step, all the words that can be stemmed to their root words are selected and

are further stemmed. The stemmed words are then applied to the TFIDF and CV concept which leads to the creation of vector tables. This table is responsible to check the presence of spam and ham mails based on ML classifiers so used. Finally, all the spam mails are rejected and the legit mails are sent to the user's inbox.

6 Experimental Analysis

To implement the functionality of the thesis, the model was tested over a corpse of dataset that were categorized as spam and ham. Using this dataset, ML models were further trained so that they could automatically categorize spam and non-spam mails. The entire classification took place on two feature sets of TFI-DF and CV. The ratio of training and testing sets were initialized as 80:20. The dataset contained a total of 2332 ham files and 1083 spam files. After the data processing that takes place messages are further split into individual words and are tokenized. This tokenized data is converted into vectors and four ML algorithms are implemented on it. Later, ML algorithms are used as classifiers and implemented using two feature sets. Finally, both the feature sets, using four ML algorithms are individually evaluated on the basis on ROC curves and precision factors. Lastly, both the feature sets are compared and the one with highest accuracy is chosen.

6.1 Simulations

The dataset undergoes multiple NLP concepts of tokenization and stemming procedures. This process is generally carried out to extract features using TFI-DF and CV concepts. Next, the model is trained with junk and legit mails and tested to classify further spam mails. The testing is done on the following evaluation metrics.

- F measure: weighted average precision and recall
- Recall: percentage of spam mails that were blocked
- Precision: percentage of spam mails

Below mentioned are its formulas that are used to calculate the above mentioned evaluation metrics:

$$Acc = \frac{TN + TP}{TP + FN + FP + TN}$$

$$F = \frac{2PR}{P + R}$$

$$R = \frac{TP}{TP + FN}$$

$$P = \frac{TP}{TP + FP}$$

6.2 Training and Testing

The 80% of the total labelled dataset is first used for training the system based on which it learns how to classify the mails that will be fetched to it for the purpose of testing the

remaining 20% of the dataset is then fetched to the trained system and on the basis of the trained data, the system is tested for accuracy and other factors and the results of the implemented ML algorithms are presented in the section below.

7 Results

7.1 Feature Set 1: Using Count Vectorizer

7.1.1 Naiive Bayes for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy of Naiive Bayes classification achieved for training and testing purpose is given below:

Table 1: Train set for NB

Multinomial NB train set using CV	
Confusion Matrix	[1813,33] [14,872]
Accuracy	98.27

Table 2: Test set for NB

Multinomial NB test set using CV	
Confusion Matrix	[458,28] [13,184]
Accuracy	93.99

The output below represents the confusion matrix and its respective ROC curve:

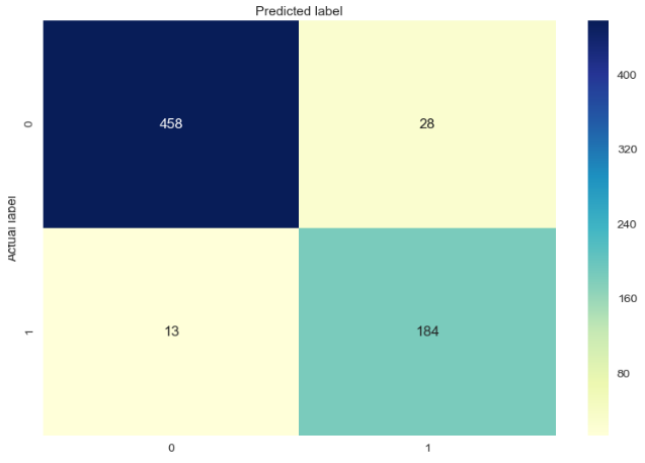

Figure 7: Confusion Matrix of NB

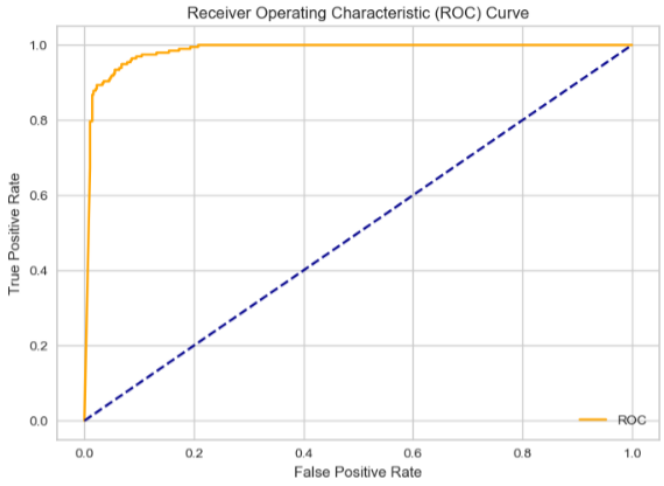

Figure 8: ROC curve of NB

As from the above figure 7, 458 cases have been predicted correctly with its respective class and 28 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 13 cases incorrectly as a class (0) of respective class (1) and 184 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 98 percent as depicted in figure 8.

Table 3: Classification Report of NB

Class Label	Precision	Recall	F1-Score	Support
Yes (1)	0.97	0.94	0.96	486
No (0)	0.87	0.93	0.90	197
Accuracy	**0.94**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Naiive Bayes is 0.94. The report below indicates the classification report of the same:

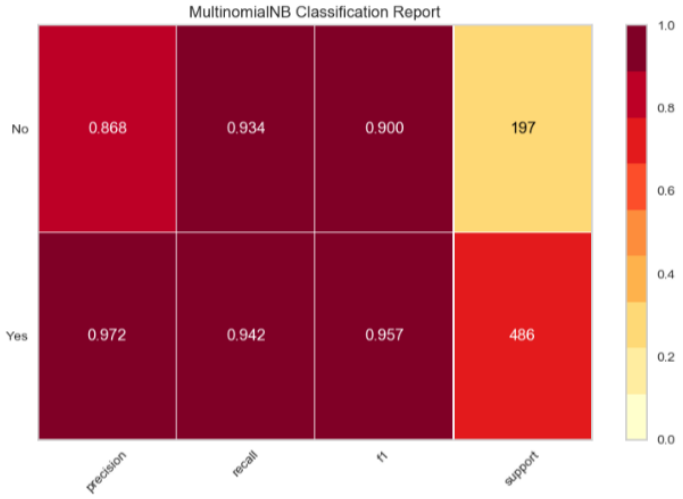

Figure 9: Classification Report of NB

The value of precision that has detected the presence of spam file is 0.97 and that of ham file is 0.87. Thus the overall precision value is 0.94, and the recall value is 0.94 and 0.93 for attack file and normal file respectively.

7.1.2 Logistic Regression for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for logistic regression classification achieved for training and testing purpose is given below:

Table 4: Train set for LR

Logistic Regression train set using CV	
Confusion Matrix	[1846,0] [0,886]
Accuracy	100.00

Table 5: Test set for LR

Logistic Regression test set using CV	
Confusion Matrix	[481,5] [8,189]
Accuracy	98.09

The output below represents the confusion matrix and its respective ROC curve:

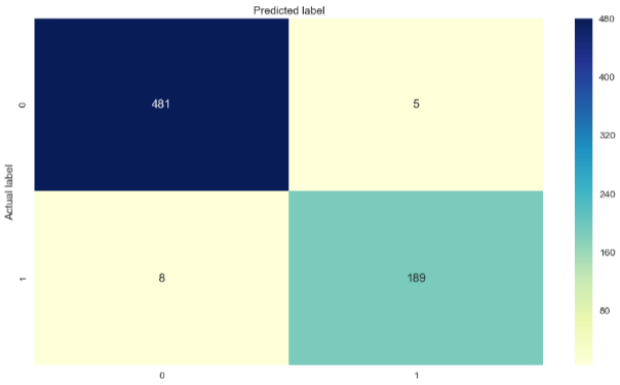

Figure 10: Confusion Matrix of LR

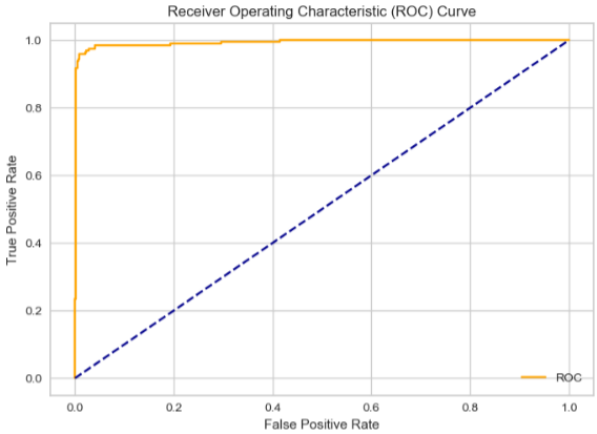

Figure 11: ROC curve of LR

As from the above figure 10, 481 cases have been predicted correctly with its respective class and 5 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 8 cases incorrectly as a class (0) of respective class (1) and 189 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 11.

Table 6: Classification Report of LR

Class Label	Precision	Recall	F1-Score	Support
Yes (1)	0.98	0.99	0.99	486
No (0)	0.92	0.96	0.97	197
Accuracy	**0.98**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Logistic Regression is 0.98. The report below indicates the classification report of the same:

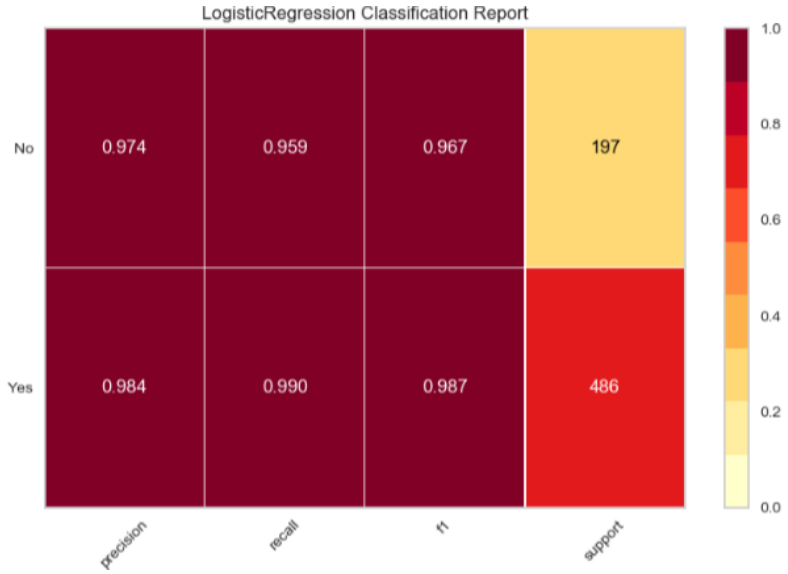

Figure 12: Classification Report of LR

The value of precision that has detected the presence of spam file is 0.98 and that of ham file is 0.92. Thus the overall precision value is 0.98, and the recall value is 0.99 and 0.96 for attack file and normal file respectively.

7.1.3 AdaBoost for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for AdaBoost classification achieved for training and testing purpose is given below:

Table 7: Train set for AdaBoost

AdaBoost train set using CV	
Confusion Matrix	[1807,39] [37,849]
Accuracy	97.2

Table 8: Test set for AdaBoost

AdaBoost test set using CV	
Confusion Matrix	[471,15] [13,184]
Accuracy	95.90

The output below represents the confusion matrix and its respective ROC curve:

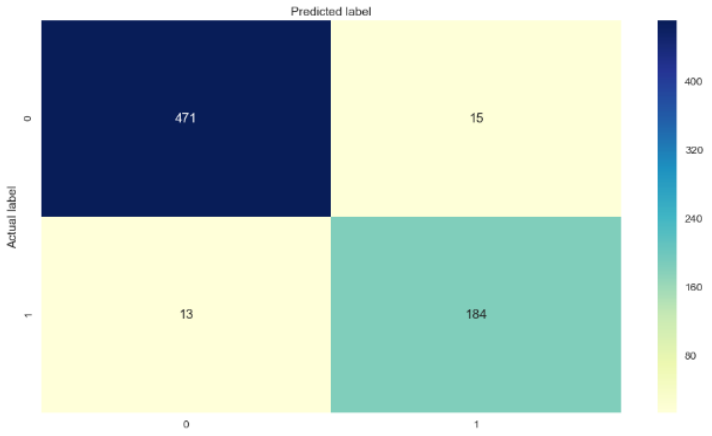

Figure 13: Confusion Matrix of AdaBoost

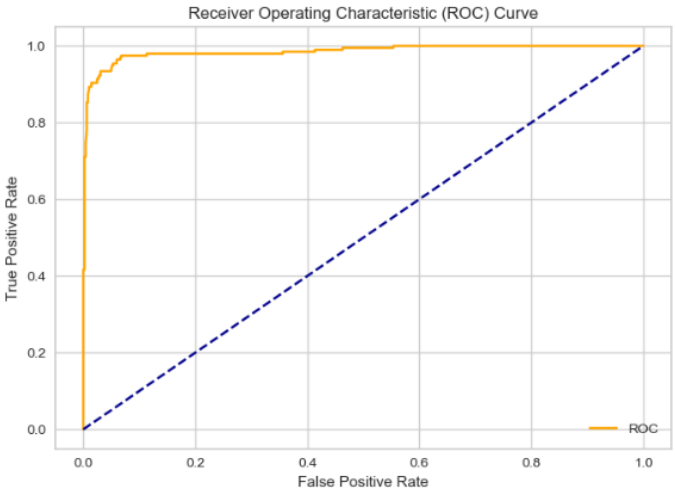

Figure 14: ROC curve of AdaBoost

As from the above figure 13, 471 cases have been predicted correctly with its respective class and 15 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 13 cases incorrectly as a class (0) of respective class (1) and 184 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 14.

Table 9: Classification Report of AdaBoost

Class Label	**Precision**	**Recall**	**F1-Score**	**Support**
Yes (1)	0.97	0.97	0.97	486
No (0)	0.92	0.93	0.93	197
Accuracy	**0.96**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through AdaBoost is 0.96. The report below indicates the classification report of the same:

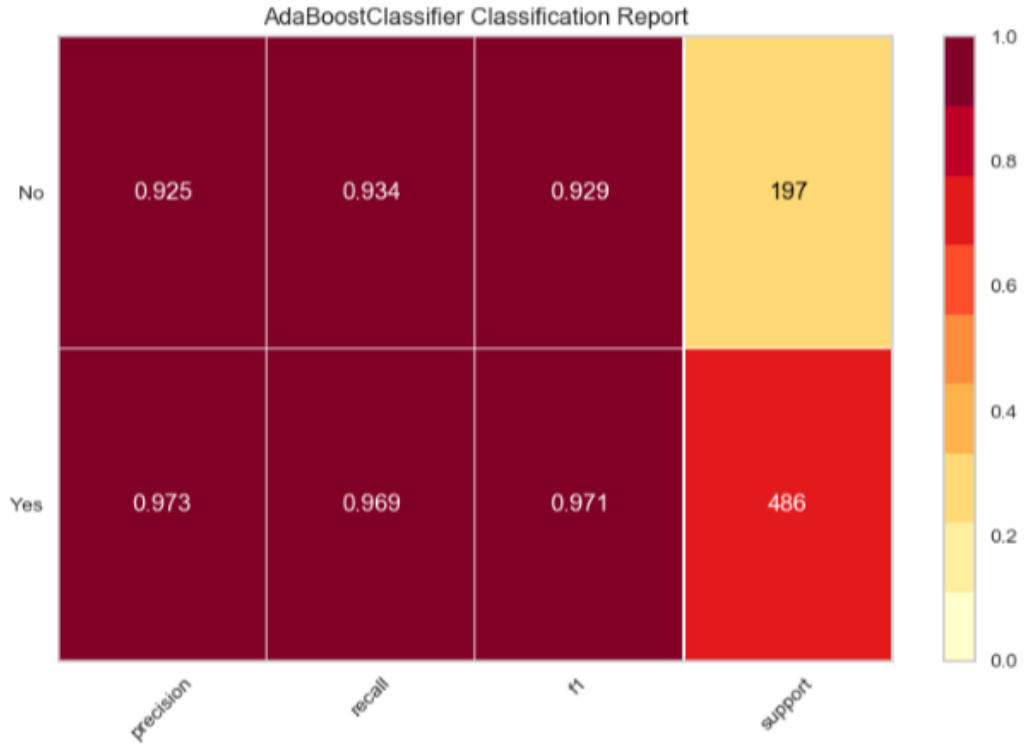

Figure 15: Classification Report of AdaBoost

The value of precision that has detected the presence of spam file is 0.97 and that of ham file is 0.92. Thus the overall precision value is 0.96, and the recall value is 0.97 and 0.93 for attack file and normal file respectively.

7.1.4 Random Forest for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for random forest classification achieved for training and testing purpose is given below:

Table 10: Train set for Random Forest

Random Forest train set using CV	
Confusion Matrix	[1846,0] [0,886]
Accuracy	100.00

Table 11: Test set for Random Forest

Random Forest test set using CV	
Confusion Matrix	[478,8] [13,184]
Accuracy	96.72

The output below represents the confusion matrix and its respective ROC curve:

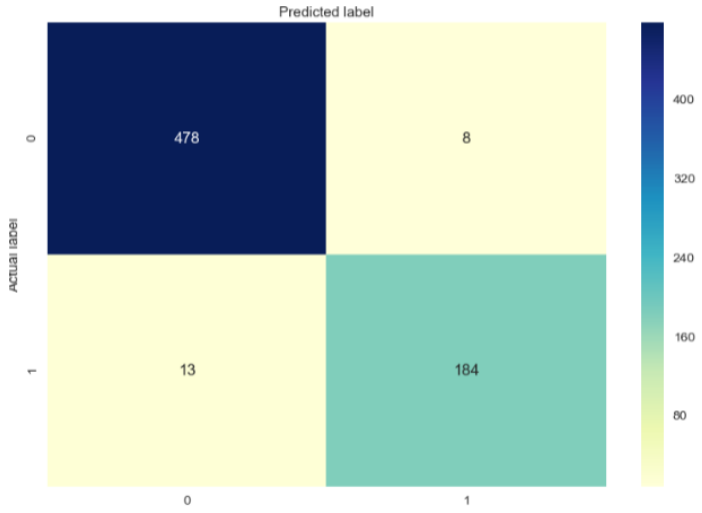

Figure 16: Confusion Matrix of Random Forest

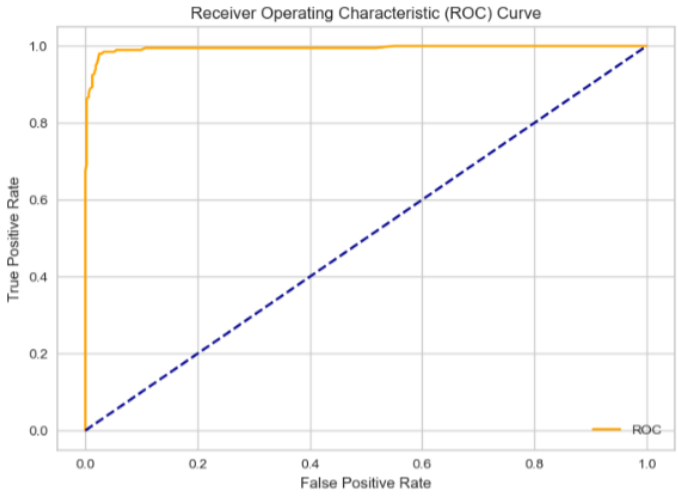

Figure 17: ROC curve of Random Forest

As from the above figure 16, 478 cases have been predicted correctly with its respective class and 8 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 13 cases incorrectly as a class (0) of respective class (1) and 184 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 17

Table 12: Classification Report of Random Forest

Class Label	Precision	Recall	F1-Score	Support
Yes (1)	0.97	0.97	0.98	486
No (0)	0.96	0.93	0.95	197
Accuracy	**0.97**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Random Forest is 0.97. The report below indicates the classification report of the same:

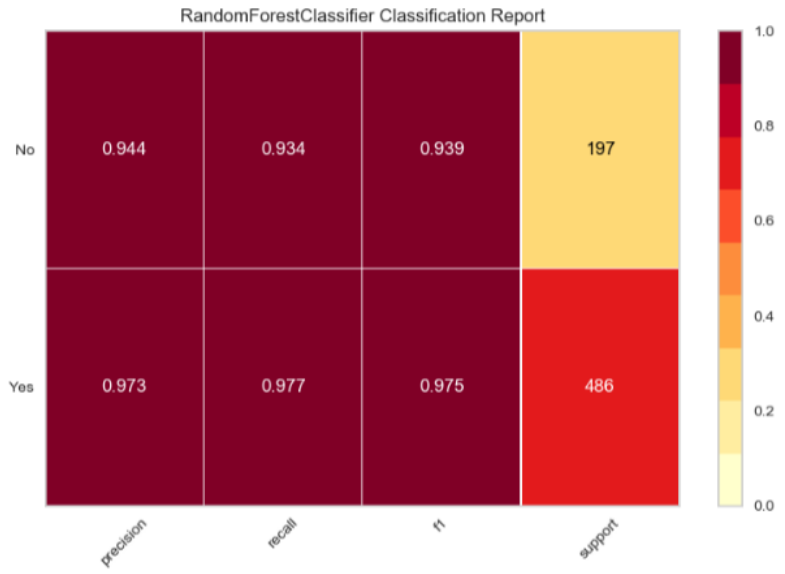

Figure 18: Classification Report of Random Forest

The value of precision that has detected the presence of spam file is 0.97 and that of ham file is 0.96. Thus the overall precision value is 0.96, and the recall value is 0.97 and 0.93 for attack file and normal file respectively.

7.2 Feature Set 2: Using TFI-DF

7.2.1 Naiive Bayes for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy of Naiive Bayes classification achieved for training and testing purpose is given below:

Table 13: Train set for NB

Multinomial NB train set using TFI-DF	
Confusion Matrix	[1846,0] [170,716]
Accuracy	93.77

Table 14: Test set for NB

Multinomial NB test set using TFI-DF	
Confusion Matrix	[486,0] [55,142]
Accuracy	91.94

The output below represents the confusion matrix and its respective ROC curve:

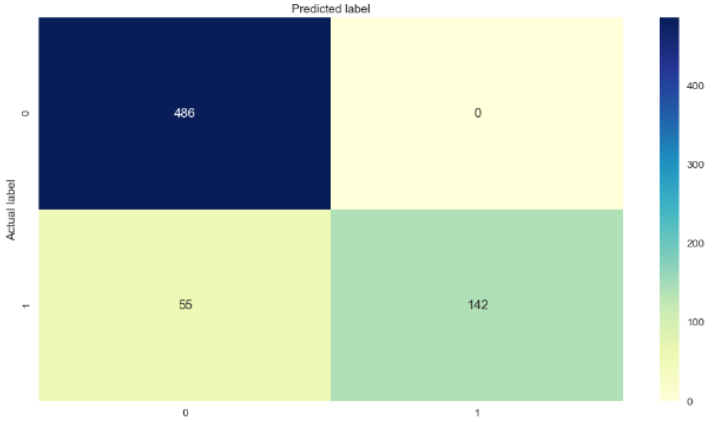

Figure 19: Confusion Matrix of NB

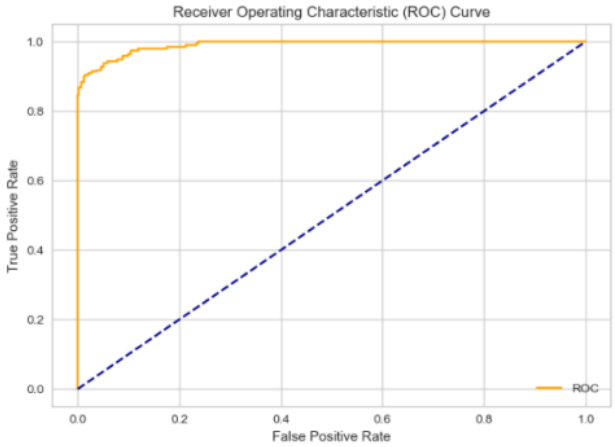

Figure 20: ROC curve of NB

As from the above figure 19, 486 cases have been predicted correctly with its respective class and 0 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 55 cases incorrectly as a class (0) of respective class (1) and 142 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 20.

Table 15: Classification Report of NB

Class Label	Precision	Recall	F1-Score	Support
Yes (1)	0.90	1.00	0.95	486
No (0)	1.00	0.72	0.84	197
Accuracy	**0.92**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Naiive Bayes is 0.94. The report below indicates the classification report of the same:

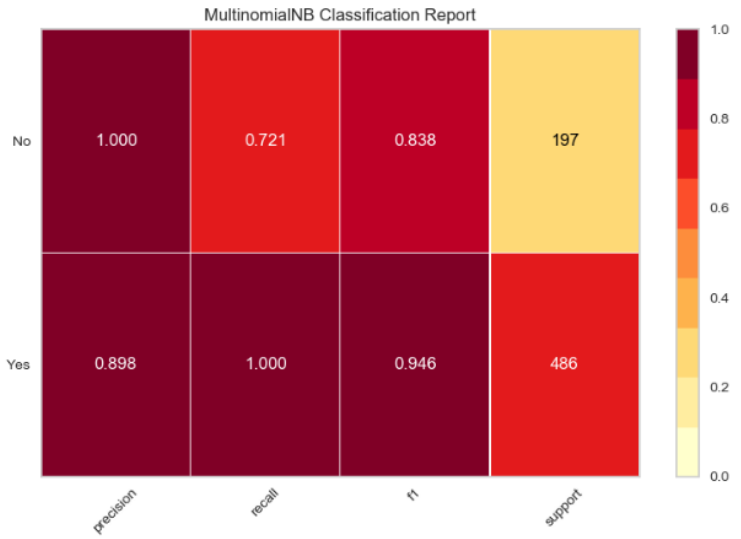

Figure 21: Classification Report of NB

The value of precision that has detected the presence of spam file is 0.90 and that of ham file is 1.00. Thus the overall precision value is 0.94, and the recall value is 1.00 and 0.72 for attack file and normal file respectively.

7.2.2 Logistic Regression for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for logistic regression classification achieved for training and testing purpose is given below:

Table 16: Train set for LR

Logistic Regression train set using TFI-DF	
Confusion Matrix	[1836,10] [50,836]
Accuracy	97.80

Table 17: Test set for LR

Logistic Regression test set using TFI-DF	
Confusion Matrix	[481,5] [32,165]
Accuracy	94.58

The output below represents the confusion matrix and its respective ROC curve:

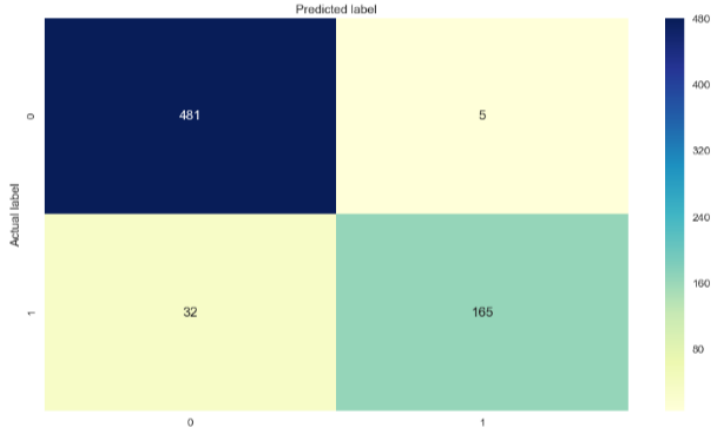

Figure 22: Confusion Matrix of LR

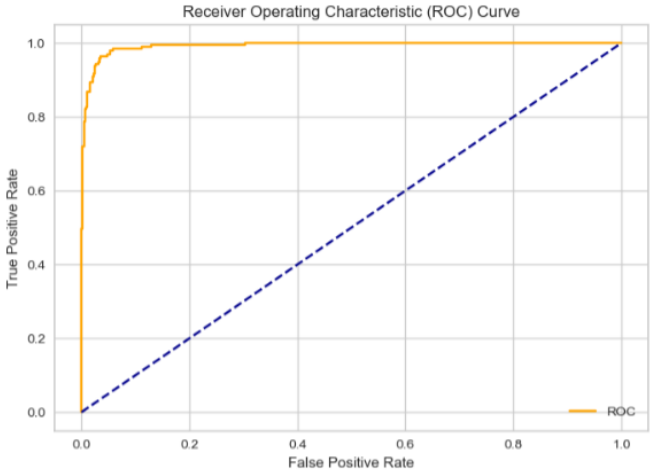

Figure 23: ROC curve of LR

As from the above figure 22, 481 cases have been predicted correctly with its respective class and 5 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 32 cases incorrectly as a class (0) of respective class (1) and 165 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 23.

Table 18: Classification Report of LR

Class Label	**Precision**	**Recall**	**F1-Score**	**Support**
Yes (1)	0.94	0.99	0.96	486
No (0)	0.97	0.84	0.90	197
Accuracy	**0.95**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Logistic Regression is 0.95. The report below indicates the classification report of the same:

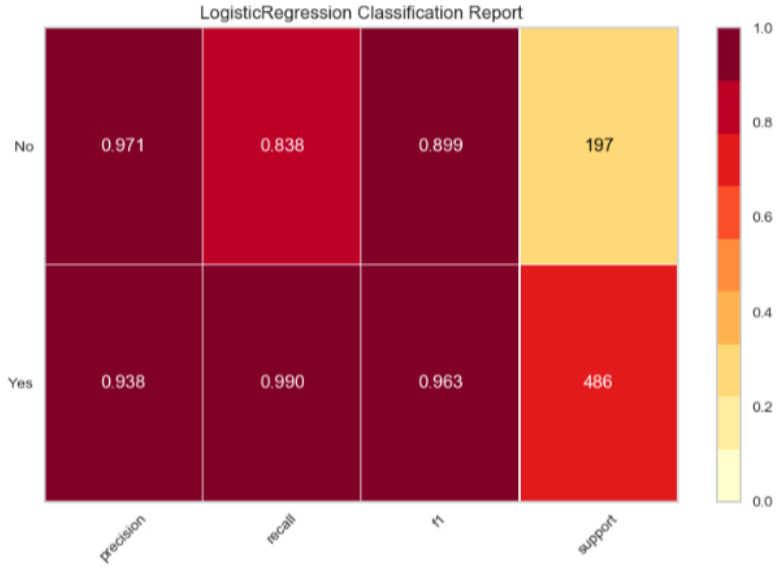

Figure 24: Classification Report of LR

The value of precision that has detected the presence of spam file is 0.94 and that of ham file is 0.97. Thus the overall precision value is 0.95, and the recall value is 0.99 and 0.84 for attack file and normal file respectively.

7.2.3 AdaBoost for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for AdaBoost classification achieved for training and testing purpose is given below:

Table 19: Train set for AdaBoost

AdaBoost train set using TFI-DF	
Confusion Matrix	[1822,24] [23,863]
Accuracy	98.72

Table 20: Test set for AdaBoost

AdaBoost test set using TFI-DF	
Confusion Matrix	[476,10] [12,185]
Accuracy	96.77

The output below represents the confusion matrix and its respective ROC curve:

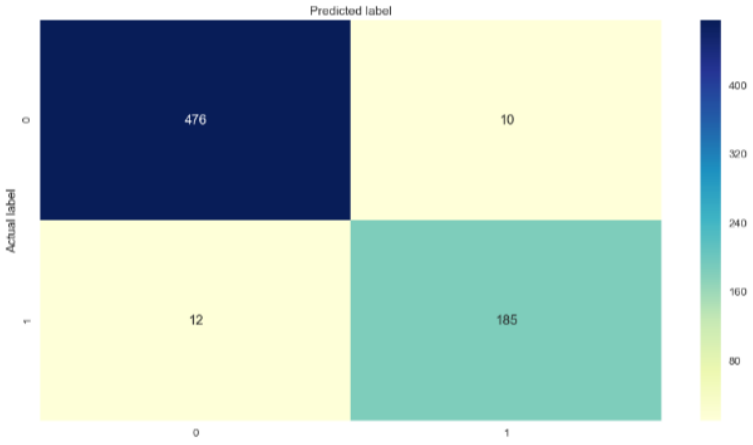

Figure 25: Confusion Matrix of AdaBoost

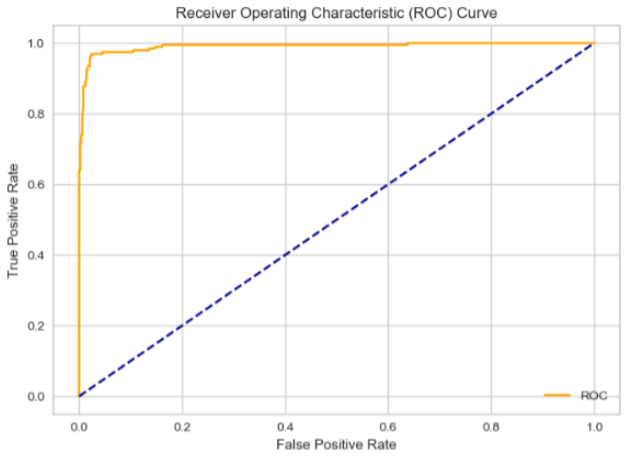

Figure 26: ROC curve of AdaBoost

As from the above figure 25, 476 cases have been predicted correctly with its respective class and 10 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 12 cases incorrectly as a class (0) of respective class (1) and 185 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 26.

Table 21: Classification Report of AdaBoost

Class Label	Precision	Recall	F1-Score	Support
Yes (1)	0.98	0.98	0.98	486
No (0)	0.95	0.94	0.94	197
Accuracy	**0.97**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through AdaBoost is 0.97. The report below indicates the classification report of the same:

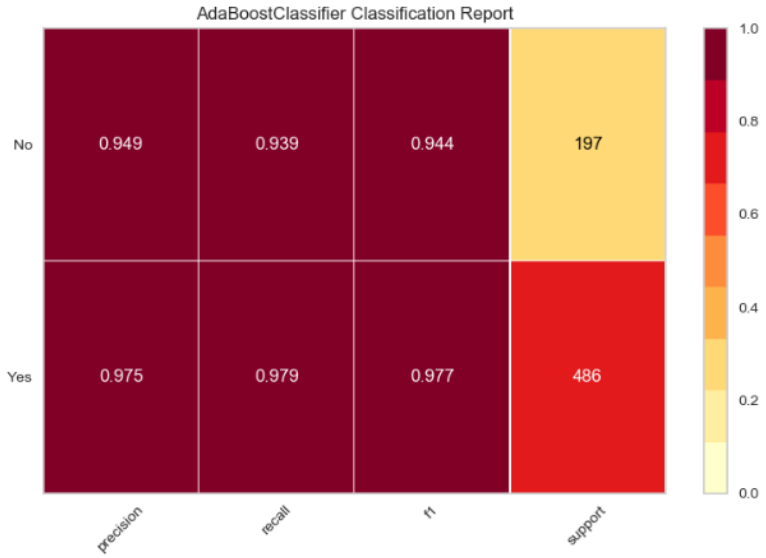

Figure 27: Classification Report of AdaBoost

The value of precision that has detected the presence of spam file is 0.98 and that of ham file is 0.95. Thus the overall precision value is 0.97, and the recall value is 0.98 and 0.94 for attack file and normal file respectively.

7.2.4 Random Forest for Spam Classification

Since the dataset is divided into train and test phases; a total of 80 percent of the dataset is used for training purpose and 20 percent for testing purpose. The accuracy for random forest classification achieved for training and testing purpose is given below:

Table 22: Train set for Random Forest

Random Forest train set using TFI-DF	
Confusion Matrix	[1846,0] [0,886]
Accuracy	100.00

Table 23: Test set for Random Forest

Random Forest test set using TFI-DF	
Confusion Matrix	[478,8] [13,184]
Accuracy	96.72

The output below represents the confusion matrix and its respective ROC curve:

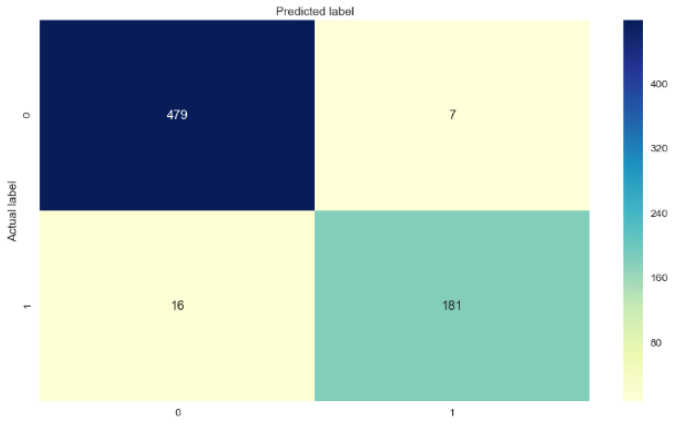

Figure 28: Confusion Matrix of Random Forest

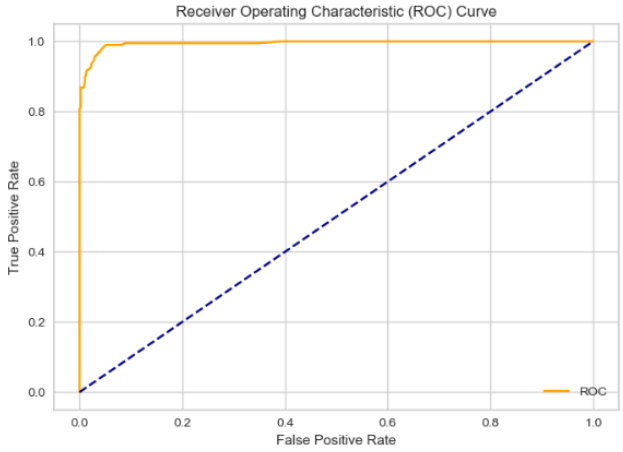

Figure 29: ROC curve of Random Forest

As from the above figure 28, 479 cases have been predicted correctly with its respective class and 7 cases are predicted wrongly as a class (1) of respective class (0). Our classifier has predicted 16 cases incorrectly as a class (0) of respective class (1) and 181 cases correctly as a class (1).

However, the ROC curve has achieved an accuracy of 99 percent as depicted in figure 29

Table 24: Classification Report of Random Forest

Class Label	**Precision**	**Recall**	**F1-Score**	**Support**
Yes (1)	0.97	0.99	0.98	486
No (0)	0.96	0.92	0.94	197
Accuracy	**0.97**			

The above table has been calculated using the confusion matrix. It has been observed that the accuracy obtained through Random Forest is 0.97. The report below indicates the classification report of the same:

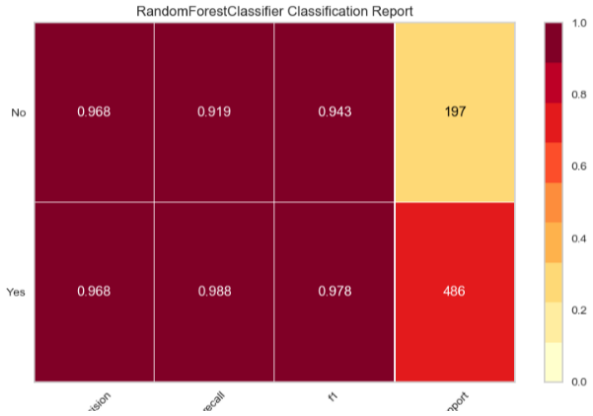

Figure 30: Classification Report of Random Forest

The value of precision that has detected the presence of spam file is 0.97 and that of ham file is 0.96. Thus the overall precision value is 0.97, and the recall value is 0.99 and 0.92 for attack file and normal file respectively

8 Conclusions

The primary aim of the thesis was to perform spam detection using ML algorithms. A total of four ML algorithms were used along with the concepts of TFI-DF and CV. On implementation it was witnessed that the model used with feature set 2 of TFI-DF performed better than the feature set 1 model of CV.

References

1. N. Govil and Astha Varshney, 2020, “A Machine Learning based Spam Detection Mechanism”,IEEE, pp.954-957
2. B. Rawat Danda and Amani Alzahrani, 2019, “Comparative Study of Machine Learning Algorithms for SMS Spam Detection”,
3. Hezha M. Tareq Abdulhadi and Cihan Varol, 2019, “Comparison of String Matching Algorithms on Spam Email Detection”, pp.6-11.
4. D. Karthika Renuka, Lakshmi Surya P, 2018, “Spam Classification Based on Supervised Learning Using Machine Learning Techniques”
5. Ghulam Mujtaba, N.Majeed, 2017, “Email Classification Research Trends: Review and Open Issues”, pp.1-21
6. Ni Zhang, Yu Jiang, Binxing Fang, Xueqi Cheng and Li Guo, “Traffic Classification-Based Spam Filter,” IEEE International Conference on Communications, vol. 5, p. 2130 – 2135, 2006.
7. Youn, Seongwook, and Dennis McLeod, “A Comparative Study for Email Classification,” Editor. Khaled Elleithy, Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 387-391, 2007.
8. Y. Zhang, R. Jin, and Z. Zhou, “Understanding bag-ofwords model:A statistical framework,” Int. J. Mach. Learn. Cybern., vol. 1, p. 43–52, 2010.
9. C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P.G. Bringas, “Study on the effectiveness of anomaly detection for spam filtering,” Inf. Sci., vol. 277, pp. 421-444, 2014.
10. E.P. Sanz, J.M.G. Hidalgo, J.C.C. Pérez, “Email spam filtering Adv. Comput,” no. 74, pp. 11-45, 2008.