A Thesis Report on

Email Spam Classification

Course Title: Project and Thesis Sessional
Course Code: CSE 402

Submitted by

Md Masum Rana
Student ID: 1902005
Level- 4, Semester- l
Dept. of Computer Science and Engineering

Hasi Rani Roy
Student ID: 1902031
Level- 4, Semester- l
Dept. of Computer Science and Engineering

Mostakim Ara Jaba
Student ID: 1802048
Level- 4, Semester- l
Dept. of Computer Science and Engineering

Department of Computer Science and Engineering,
Hajee Mohammad Danesh Science & Technology University, Dinajpur-5200

Department of Computer Science and Engineering
Faculty of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University Dinajpur-5200, Bangladesh

CERTIFICATE

This is to certify that the work entitled as "**Email Spam Classification"** by Md. Masum Rana, Hasi Rani Roy and Mostakim Ara Jaba has been carried out under my supervision. To the best of my knowledge this work is an original one and was not submitted anywhere for a diploma or a degree.

Supervisor

……
(Md. Fazle Rabbi)
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200, Bangladesh

Department of Computer Science and Engineering
Faculty of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University
Dinajpur-5200, Bangladesh

DECLARATION

The work entitled "**Email Spam Classification**" has been carried out in the Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University is original and conforms to the regulations of this University.

We understand the University's policy on plagiarism and declare that no part of this thesis has been copied from other sources or been previously submitted elsewhere for the award of any degree or diploma.

……
(Md. Masum Rana)
Student ID: 1902005
Session: 2019

……..
(Hasi Rani Roy)
Student ID: 1902031
Session: 2019

……
(Mostakim Ara Jaba)
Student ID: 1802048
Session: 2018

Acknowledgment

We would like to express our thanks of gratitude to Md. Fazle Rabbi Professor, department of computer science and engineering who gave us a golden opportunity to do this thesis and also provided support in completing in our thesis. His heartiest & kind Cooperation during our thesis work makes the dream real & we succeed to complete our thesis.

While we were preparing this thesis file, various information that we found helped us in chapter of profile adding and we are glad that we were able to complete this thesis and understand many things. Through preparation of **Email Spam Classification** thesis paper was an immense learning experience and we inculcated many personal qualities during this process like responsibility, punctuality, confidence and others.

We would like to thank to our supervisor who supported us all the time, cleared our doubts and to our parents who also played a big role in finalization of our thesis file. We are taking this opportunity to acknowledge their support and we wish that they keep supporting us like this in the future. A thesis work is a bridge between theoretical and practical learning and with this thinking we worked on the thesis and made it successful due to timely support and efforts of all who helped us.

Contents

List of Figures

List of Tables

Abstract

E-mail is one of the most popular and frequently used ways of communication due to its worldwide accessibility, relatively fast message transfer, easy to use and low sending cost. For these advantages, most of the institutions and companies prefer to use emails over all other mediums. But nowadays, spam email has become a major problem, with rapid growth of internet users, spam email is also increasing. People are using them for illegal and unethical conducts and fraud. Sending malicious link through spam emails which can harm our system. Creating a fake profile and email account is much easy for the spammers, they pretend like a genuine person in their spam emails, these spammers target those peoples who are not aware about these frauds [1] .So, it is needed to identify those spam mails which are fraud, this project will identify those spam by using techniques of machine learning. This paper will discuss the machine learning algorithms and apply all these algorithms on our data sets and best algorithm is selected for the email spam detection having best precision and accuracy.

Keywords: Machine learning, Naïve Bayes, Support Vector Machine, K-nearest Neighbor, Decision tree, Logistic regression.

Chapter 1
Introduction

1.1 Introduction:

Email or electronic mail spam refers to the "using of email to send unsolicited emails or advertising emails to a group of recipients. Unsolicited emails mean the recipient has not granted permission for receiving those emails. According to Wikipedia "the use of electronic messaging systems to send unsolicited bulk messages, especially mass advertisement, malicious links etc." are called as spam. "Ham" this term was given by Spam Bayes around 2001 and it is defined as "Emails that is not considered spam" [7]. Spam is a waste of storage, time and message speed. The problem of spam e-mail has been increasing for years. In recent statistics, 55.5% of all emails are spam which about 120+ billion email per day. Automatic email filtering may be the most effective method of detecting spam but nowadays spammers can easily bypass all these spam filtering applications easily. Several years ago, most of the spam can be blocked manually coming from certain email addresses. Spammers began to use several tricky methods to overcome the filtering methods like using random sender addresses and/or append random characters to the beginning or the end of the message subject line [2]. Many researchers are already working on spam filtering techniques, but accurate spam detection is considered a difficult task due to many reasons including subjective nature of spam, processing overhead and message delay, type of language used and irregular cost of filtering errors. Text mining approach is used for the classification of mail as spam and non spam. Different machine learning algorithms have been used by different authors for the detection and classification of spam mails [3], discussed in chapter 2.

Machine learning algorithm does not require specifying any rules. Instead, a set of training samples, these samples is a set of pre classified e-mail messages. A specific algorithm is then used to learn the classification rules from these e-mail messages. Machine learning approach has been widely studied and there are lots of algorithms can be used in e-mail filtering. They include Naïve Bayes, Support Vector Machines, K-nearest neighbor, Logistic regression, Decision tree.

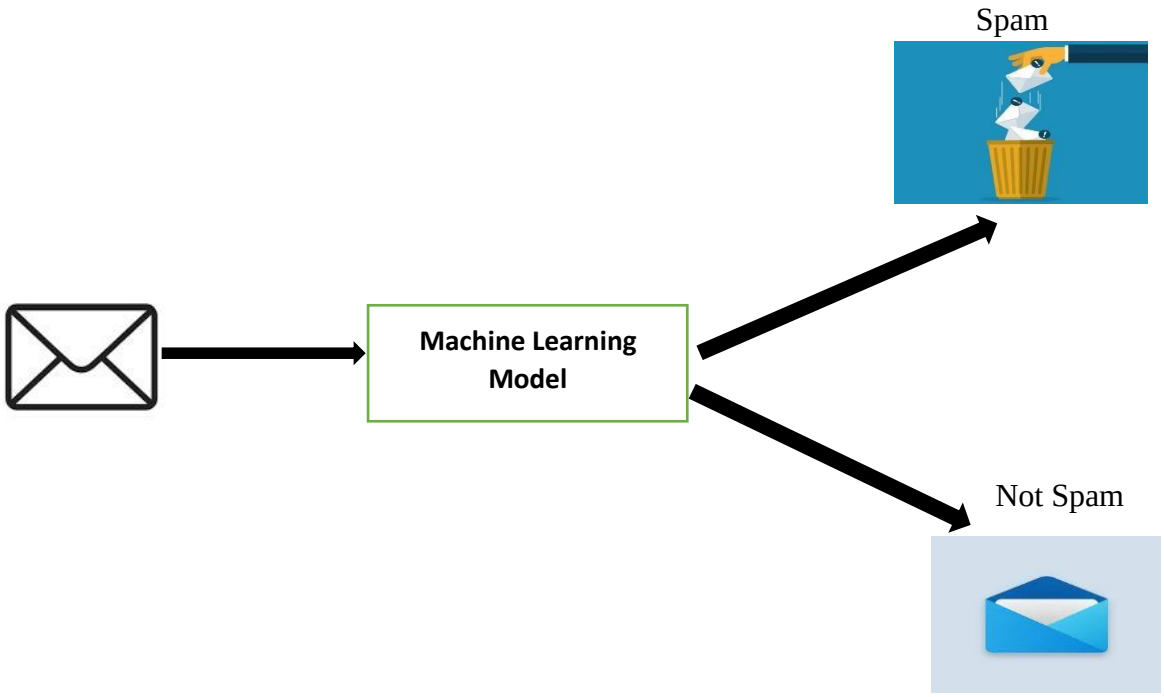

Fig.1.1 Classification into Spam and non-spam

In this paper, we propose a framework for spam email detection using an unified approach of machine learning based Naïve Bayes algorithm and we also explain Support Vector Machines, K-nearest neighbor, Logistic regression, Decision tree to compare with our proposed model. Naive Bayes classifiers are simple probabilistic classifiers based on Bayes' theorem. They are popular for text classification tasks like spam filtering. SVMs are effective for binary classification tasks like spam filtering and DT is also used for spam classification.

1.2 Motivation & Inspiration:

In today's digital age, where email communication serves as a fundamental tool for personal and professional interactions, the prevalence of spam emails poses significant challenges for users worldwide. With the exponential increase in spam emails, there is an urgent need for robust and efficient spam filtering mechanisms to safeguard users' inboxes and ensure a seamless communication experience.

The motivation and inspiration for an email spam classification project can stem from several factors:

User Experience Improvement: Enhancing the user experience by reducing the annoyance and inconvenience caused by spam emails, thus improving productivity and efficiency.

Security Enhancement: Protecting users from potentially harmful content such as phishing scams, malware, or fraudulent schemes often propagated through spam emails.

Resource Optimization: Minimizing the resources wasted on processing and managing spam emails, including storage, network bandwidth, and computational power.

Learning Opportunity: The project provides a valuable learning experience in machine learning, natural language processing, and data analysis techniques.

1.3 Objective:

The primary objective of the email spam detection project is to develop a robust and efficient system capable of accurately identifying and filtering out spam emails from legitimate ones. This involves employing advanced machine learning and natural language processing techniques to analyze email content, and other relevant features to differentiate between spam and non-spam emails. The key goals of the project include:

- To effectively identify and filter out spam emails from users' inboxes, thereby reducing the risk of users falling victim to phishing scams, malware, or other malicious activities associated with spam.
- Develop a spam detection model with a high level of accuracy to minimize false positives (misclassifying legitimate emails as spam) and false negatives (allowing spam emails to reach the inbox).
- Also increase precision value to predict spam emails appropriately
- Design the system to be scalable, capable of handling large volumes of emails efficiently without compromising performance.
- To extract features from a text-based dataset for further analysis of the spamming traits.

1.4 Problem Statement:

Email is one of the platforms which play a vital role in communication of useful information. With the study of several datasets by different researchers, they are successfully able to predict spam for emails. Though several researches have been done in the field, still a vast area is yet to cover up. Previous studies have shown promising results in detecting spam in email through the analysis of several machine learning algorithms Naive Bayes and Decision tree [7]. However, these studies have also highlighted the need for further research to improve the accuracy and reliability of existing algorithms. Detecting spam through textual data from email is challenging but can be improved with more accurate algorithms and approaches. While there has been

previous research in this field, there is still much more to explore, and updated datasets can enhance accuracy and performance. Keeping this in mind, the following questions need to be addressed:

- What are the challenges in implementing text-based spam detection?
- Which Feature Extraction methods are more effective for detecting spam?
- What is the most efficient machine learning algorithm for detecting spam?

1.5 Contribution:

In this paper, we consider different machine learning algorithms for spam detection. Our contributions are delineated as follows:

i. The study discusses various machine learning based spam filters, their architecture, along with their pros and cons. We also discussed the basic features of spam email.
ii. Some exciting research gaps were found in the spam detection and filtering domain by conducting a comprehensive survey of the proposed techniques and spam's nature.
iii. Open research problems and future research directions are discussed to enhance email security and filtration of spam emails by using machine learning methods.
iv. Several challenges currently faced by spam filtering models and the effects of those challenges on the models' efficiency are discussed in this study.
v. A comprehensive comparison of machine learning techniques and concepts that help understand machine learning's role in spam detection is provided.

1.6 Boundary of the Work:

The analysis primarily focuses on classifying spam messages using machine learning techniques applied to a specific dataset. It follows a structured approach encompassing data preprocessing, natural language processing (NLP), feature extraction, model building, and performance evaluation. The study is bounded by the scope of the dataset used. Additionally, the analysis is limited to the algorithms, feature engineering techniques, and ensemble learning methods implemented within the chosen framework.

1.7 Thesis Organization:

This thesis is structured into five chapters, each with a specific focus. A brief summary of these chapters is presented below.

Chapter 1 serves as an introduction to this thesis, highlighting its main objective, motivation and inspiration, boundary of the works.
Chapter 2 provides an overview of the existing research on the topic, highlighting key themes, trends, and gaps in the literature.
Chapter 3 outlines the complete methodology proposed in this thesis. It covers various aspects such as dataset description, preprocessing techniques, feature encoding and selection methods, performance measures, and the models employed.
Chapter 4 delves into the conducted experiments and presents detailed findings. It critically evaluates and discusses the outcomes of these experiments.
Chapter 5 concludes the work presented in this thesis and outlines potential directions for future research.

Chapter 2
Literature Review

2.1 Introduction

Spam email classification using machine learning models aims to provide a comprehensive overview of the current state of research in this area. The research will explore existing literature that utilizes spamming tactics and techniques such as empirical studies, reviews, and Meta analyses to classify spam email. It will examine different methods and approaches employed in classifying spam email, including Machine learning (ML) algorithms, Natural Language Processing (NLP) techniques, feature extraction methods, and feature engineering. The discussion will encompass the accuracy, reliability, and validity of spam email classification; ML techniques have gained significant popularity in the classification of spam email due to their efficiency and scalability. This literature overview will highlight recent publications that utilize ML and NLP approaches to classify spam email in various domains.

2.2 Background Study

In the past several years, there has been a growing interest in utilizing machine learning and deep learning techniques for classifying spam email messages. Researchers have explored various approaches and methodologies to leverage text data for accurate and early identification of whether a message is spam or not. Some of them are briefly described below.

2.2.1 Spam Email Classification using Machine Learning

Spam Email Classification using Machine Learning Supervised machine learning methods have been utilized to address the problem of spam email tactics, which includes techniques such as logistic regression, support Vector machine, naive bayes and ensemble learning methods such as random forests and XGBoost.

2.2.2 Spam Email Classification using Deep learning

Spam Email Classification using Deep learning is a type of machine learning method that employs multiple non-Logistic layers to convert low-level data representations into higher-level ones. This technique is commonly utilized for analyzing, extracting features, classifying, and identifying patterns. Popular deep learning architectures include feed-forward neural networks, convolution neural networks, and recurrent neural networks. In the realism of natural language processing, various deep learning models such as ANN, CNN, and this produce have produced favorable results.

2.3 Related Works

Email spam is nothing more than fake or unwanted bulk mails sent via any account or an automated system. Spam emails are increasing day by day, and it has become a common problem over the last decade. Email IDs receiving spam emails are typically collected through spam bots. The applications of machine learning have been playing a vital role in the detection of spam

emails. It has various models and techniques that researchers are using to develop novel spam detection and filtering models [4].

Some of the techniques are able to detect both textual and image data format while some can only detect textual data format. Different strategies are tailored by totally different completely different authors with experimentation on different datasets. Some authors have worked on the detection of spam email in each the matter and image data formatting.
There is some related work that apply machine learning methods in email spam detection such as Nikhil Kumar, Sanket Sonowal, Nishant [1] They describe a focused literature survey of Machine learning methods for email spam detection. Ankita Sharma, Harshita Jain, Dr. Amol K. Kadam have used the NB and Particle Swarm Optimization techniques for the e-mail spam detection [3]. Besides R. Deepa Lakshmi and N.Radha have used methods of using Data Mining Tools [5] and W.A. Awad, S.M. ELseuofi [2] have used of SVM, NB, KNN,RS and AIS with experimentation on dataset. Either a number of the authors have used SVM separately (Renuka and Visalakshi, 2014)[7] or some have used SVM in integration with another ideas like SVM-NB (Feng et al., 2016)[8], SCS-SVM (Kumaresan and Palanisamy, 2017)[9], and SVM-ELM (Olatunji et al., 2017)[10]. In 2014, Renuka and Visalakshi have used Support vector Machine (SVM) for the classification of Email Spam detection beside the employment of Latent linguistics compart mentalization (LSI) for feature choice. TFIDF is employed for the feature extraction. Here, planned SVM-LSI is compared with SVMTFIDF while not victimization LSI, PSO and Neural Network. From the thought-about strategies, SVMLSI performs higher in terms of accuracy as compare to alternative existing ideas. In 2016, Feng et al. have planned SVM-NB algorithmic program for the e-mail spam filtration. Authors have combined the SVM algorithmic program with NB approach wherever NB will handle massive dataset and SVM is ready to make hyper- plane primarily based separation between completely different feature classes. ELM is machine learning approach that was planned to beat the perennial downside of feed forward neural network and is employed as learning approach for single layer primarily based neural network. Results of ELM and SVM as compared on the idea of Accuracy and Time taken for the e-mail spam classification from same dataset. In terms of Accuracy, SVM performs higher with 94.06 considered compare to ELM having accuracy 93.04%. Except for every case, SVM consumes longer as compare to ELM. So, ELM is healthier than SVM in terms of your time taken.

Most of the researchers have targeted solely on the text primarily based email spam classification as image based spam are often filtered at the initial stage of pre-processing.
In this paper, we work on Naïve Bayes because it gives us highest accuracy and precision than others algorithms.

Chapter 3

Methodology

3.1 Introduction:

The methodology part of this report on Email Spam Detection describes the data sources, tools, techniques, and procedures used to collect, process, and analyze text-based data from email for detecting spam using several machine learning techniques.

3.2 Proposed Methodology:

The proposed system framework contains four distinct steps: Dataset preparation, Feature Engineering, Machine learning model preparation, Evaluation criteria determination. Firstly, email data are collected and then labeled manually in a CSV file. These raw data needed to preprocess otherwise it will highly degrade the performance of classifiers. Therefore, to give the dataset a useable representation, various efficient data preparation techniques are chosen. And to extract and select the most important features from the dataset, the TF-IDF approach is considered. Afterwards, five ML algorithms (e.g., SVM, LR, DT, NB, KNN) are chosen and the study attempted to investigate which supervised machine learning model is best suited for the classification purpose.

In this chapter, ours aim is to provide an overview of the methodology and technologies used in the development of "Email spam classification" paper. This model has used email data set from a online website, Kaggle. By analyzing kaggle data to detect and classify the problems people are undergoing in the world, the study proposed a system architecture shown in Fig. 3.1

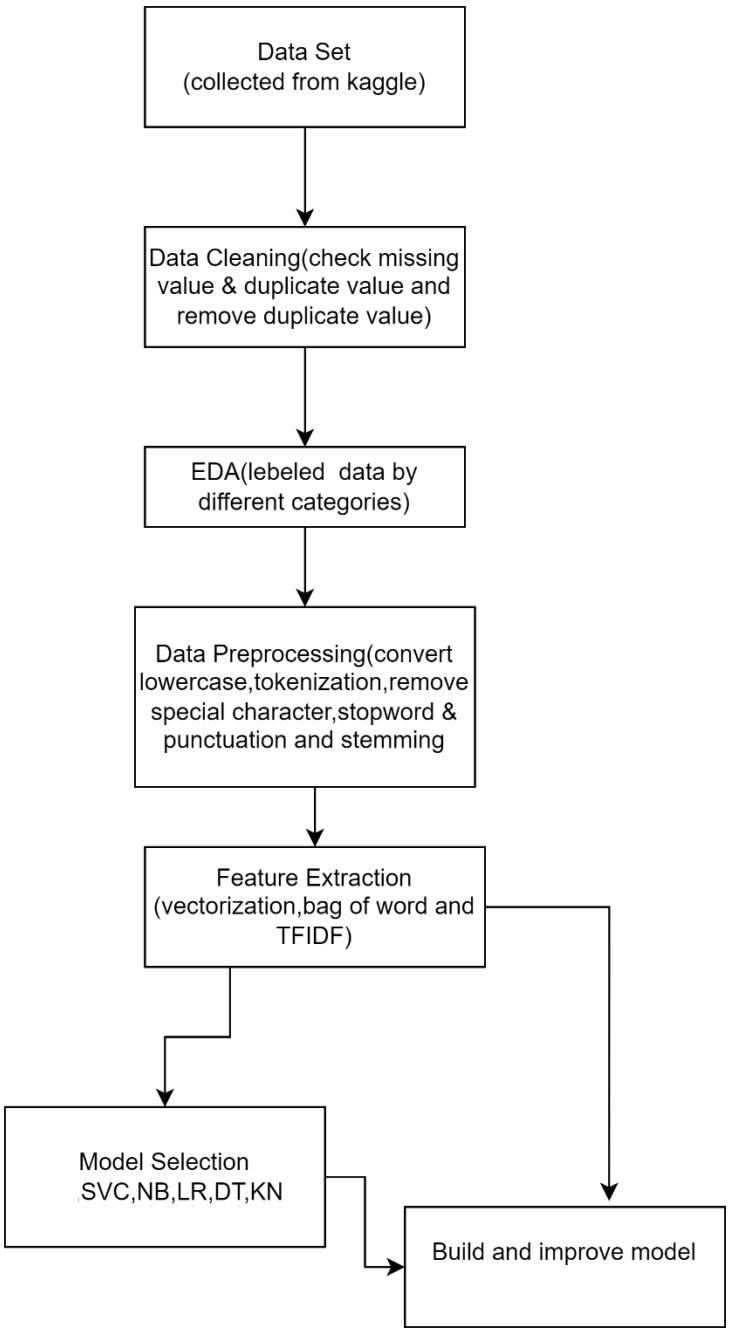

Fig. 3.1 Block Diagram of System Architecture

3.3 Data Collection

In this research, we have used only one datasets which is collected from kaggle. The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged acording being ham (legitimate) or spam. The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text [6]. A large amount of noise and irrelevant information can be found in the publicly available datasets, which can affect how well our model performs. Preprocessing datasets with many techniques is necessary to eliminate such information.

3.4 Data Cleaning

The practice of correcting or eliminating inaccurate, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are numerous ways for data to be duplicated or incorrectly labelled when integrating different data sources. In this paper, first we have to check whether some missing values are available. Later, we checked for duplicate values; if there was any duplicate value found in the message, we removed the duplicate value.

3. 5 EDA (Exploratory Data Analysis)

Exploratory Data Analysis (EDA) is a process of describing the data by means of statistical and visualization techniques in order to bring important aspects of that data into focus for further analysis. This involves inspecting the dataset from many angles, describing & summarizing it without making any assumptions about its contents. Our data set was labeled into various categories in this part. We first categorized the dataset in this case using ham and spam. Following the grouping of the dataset based on the quantity of characters, words, and phrases, a pie chart is plotted.

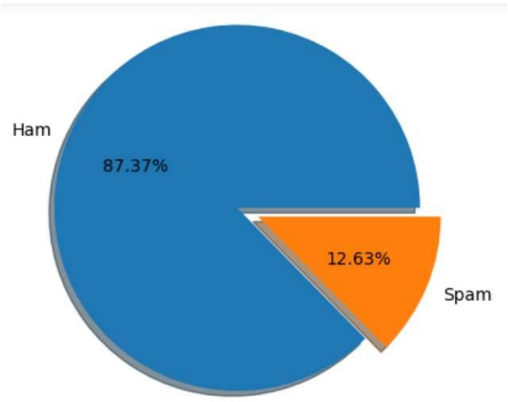

Figure-3.2: Diagrammatic representation of dataset

To gain insight into the most commonly occurring words in our dataset, we created a Word Cloud, which is a visual representation of the most frequently used words. A Word Cloud can be a valuable tool for identifying themes and patterns in the text. The size of each word in the Word Cloud is proportional to its frequency of occurrence in the dataset. By examining the Word Cloud, we can identify the most commonly used words in the dataset and gain a better understanding of the topics and themes that are discussed.

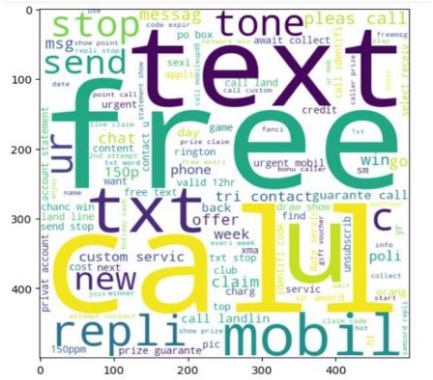

Figure-3.3: Word Cloud

In addition to the Word Cloud, we have also generated a histogram that displays the frequency of the top 30 most commonly occurring words in the dataset. The histogram provides a more detailed view of the frequency distribution of the words and can be useful in identifying specific words that are more prevalent in the dataset. The graph was created using the matplotlib and is shown below.

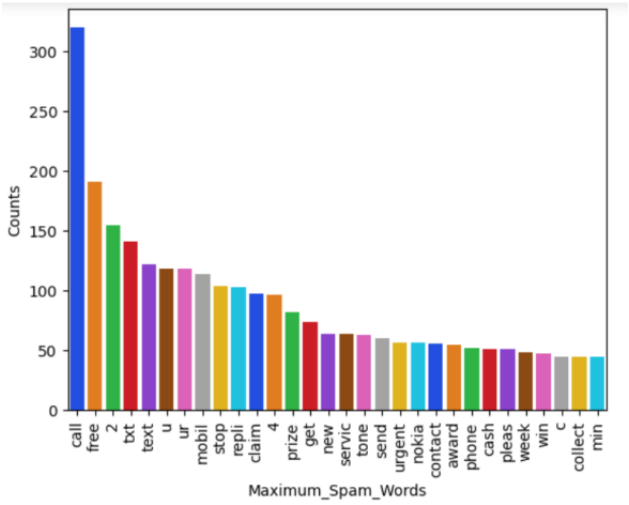

Figure-3.4: Histogram of Frequency of Words

3.6 Data preprocessing:

Textual pre-processing is crucial and significant when it comes to spam identification. The primary goal of text data preprocessing is to eliminate information that isn't helpful in identifying the document's class. Moreover, we also like to eliminate unnecessary data. The two data cleaning techniques that are most frequently employed in textual retrieval tasks are eliminating stop words and stemming to narrow the vocabulary. The data preprocessing stage in this investigation consists of two phases. The first stage involves data preparation, which includes a variety of operations such as lowercase text conversion, stop word removal, separate character removal, tokenization, and stemming. We do feature extraction and feature selection in the second stage.

3.7 Data Preparation:

The data collected from Kaggle would not be feasible for analysis. These unstructured data contain lots of noise e.g., punctuations, special characters, stop words, etc. which can negatively affect the data analysis. Therefore, it is needed to remove those unnecessary and unwanted elements from the dataset. By cleaning and manipulating the raw data, it will produce more accurate, reliable and useful information which can surely help for better performance in data analysis. For text preparation, various effective techniques which were carried out to achieve the best outcome are discussed below:

- Convert text to lowercase
- Tokenization
- Remove special characters
- Removal stop words
- Stemming

3.7.1 Convert text to lowercase:

By converting the text to lowercase such as these two words “Presentation!” and “presentation!”, would not be considered as a different word anymore. It helps in extracting the most relevant features from the dataset effectively.

3.7.2 Tokenization:

It is the process of splitting every sentence into a set of meaningful text (token). Tokens are parted by whitespaces characters, like line break or space, or by punctuation characters. Every single neighboring string of alphabetic characters are a piece of one token. White spaces and punctuations might or might not involve in the resulting lists of tokens. The tokens are very important components of the dataset and are considered as the base elements for stemming.

3.7.3 Remove special characters:

These special characters (e.g., punctuation, non-alphanumeric characters) don't have any impact on the meaning of text, therefore, they were removed for computational objectives. These bad characters were replaced with whitespace using regular expressions.

3.7.4 Removal stop words:

Stop words are the most commonly used words (e.g., "the", "a", "an", "in"), they do not provide any support to find the context or actual meaning of the text. So, the stop words were ignored.

3.7.5 Stemming:

Once we've eliminated the "noise" from our text data, it's time to normalize the data set. Different words can be used in various ways, like "stop" and "halted"(past participle) or "price" and "prices"(plural). Text normalization is the process of converting these different word forms into their root form.
To perform text normalization, we can use a tool called Word Net Lemmatizer, which is available in the NLTK package. This tool helps to transform words to their base or root form, so that we can more easily analyze and understand the data.

3.8 Feature engineering:

Feature engineering is a crucial step in the process of transforming raw data into features that can be used in supervised learning. This involves selecting, creating, and modifying features to improve the performance and understandability of the models. To accomplish this, we employed two primary feature engineering techniques in this study: feature extraction and feature selection.

- Feature Extraction.
- Feature Selection.

Feature extraction involves extracting relevant information from raw data to create new features. Feature selection, on the other hand, involves choosing the most important features from the extracted set of features to minimize complexity and improve the accuracy of the models. By utilizing these feature engineering techniques, we can generate features that are optimized for supervised learning and enable us to make more accurate predictions.

3.8.1 Feature extraction:

The important set of features from the dataset are needed to be extracted such that it can boost the overall performance of text data analysis. Feature extraction reduces the number of redundant data without losing vital information. The features can be extracted based on the weight and frequency measure of words from texts. The NLP technique Bag of words, TF-IDF vectorization were applied to extract the most important features. "Bag of Words (BOW) is a method of extracting features from text documents. Further these features can be uses for training machine

learning algorithms. Bag of Words creates a vocabulary of all the unique words present in all the document in the Training dataset.". TF-IDF vectorizer defines the statistics of how important a word is to a specific problem class compared to all the other words in the dataset [6]. In both processes, the given dataset converted into numerical feature vectors. Count vectorizer generates vector of word counts of each instance from the dataset.

3.8.2 Feature Selection:

In machine learning, we need to select the most important features from our data for optimal model performance. We use a technique called "feature selection" to achieve this. In our model, we use a specific feature selection method called "SelectFromModel", which uses an algorithm called "Extra Tree Classifier" to choose the best features for our model. This method selects the most crucial features from the dataset based on their weightage importance, which is compared to a specific value we set. We need to have a specific model to use this method, and we use the "Extra-Tree Classifier" model. The model sets a specific threshold value using a method called "median method" and fits the model with the extracted features and output. This method helps us to improve our model's performance by selecting only the most relevant and important features from our data.

3.9 Machine learning algorithms:

Machine learning algorithms are computer programs that can automatically learn and make predictions based on patterns in data. There are three main types of machine learning algorithms:

- Supervised Learning Algorithms.
- Unsupervised Learning Algorithms.
- Reinforcement Learning algorithms.

In this work, we employ a supervised learning technique to forecast our model's performance. Different algorithms can be applied to different tasks; for example, support vector machines can be used to solve classification difficulties and logistic regression to solve prediction problems. The exact algorithms we employed in our investigation are listed below:

3.9.1 Support Vector Machine (SVM):

A SVM is a supervised machine learning algorithm that can be used for classification or regression problems. It works by finding the optimal hyperplane that separates the data points of different classes with the maximum margin. The data points that lie on the margin are called support vectors, and they determine the position and orientation of the hyperplane. SVMs can handle Logistic and nonLogistic data by using different kernel functions, such as Logistic, polynomial, radial basis function (RBF), or sigmoid. SVMs are effective.

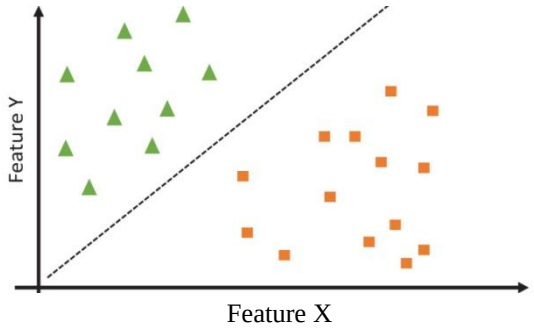

Figure-3.5: Support Vector Machine

3.9.2 Naive Bayes:

The Naive Bayes classifier is a machine learning model that uses probability to classify data. It is based on the Bayes theorem, which is at the core of the classifier's functionality. In our model we used Multinominal Na¨ıve Bayes. It is commonly utilized for document classification tasks, where the goal is to determine if a document belongs to a specific category such as sports, politics, or technology. To accomplish this, the classifier uses the frequency of words present in the document as features or predictors.

$$P(c|X) = \{P(X|c)P(c)\}/P(X)$$

3.9.3 Logistic Regression:

In the case of Logistic Regression (LR), it is not a regression but a classification algorithm. It is an algorithm that performs predictive analysis by estimating probabilities. LR uses a complex cost function which is known as 'sigmoid function' or 'logistic function' and maps prediction to a discrete set of categories. More precisely, it predicts the probability of frequency of an event by fitting data to the sigmoid function and calculates discrete values to perform prediction on a given set of independent variables.

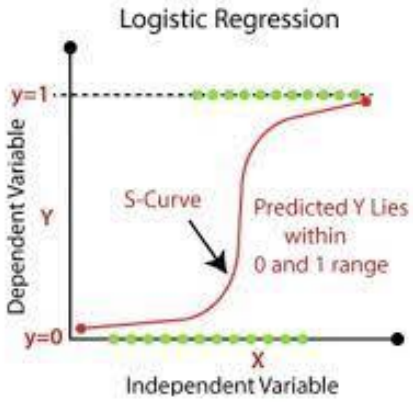

Figure-3.6: Logistic Regression

3.9.4 K-nearest neighbor classifier:

The k-nearest neighbor (K-NN) classifier is considered an example-based classifier, that means that the training documents are used for comparison rather than an explicit category representation, such as the category profiles used by other classifiers. As such, there is no real training phase. When a new document needs to be categorized, the k most similar documents (neighbors) are found and if a large enough proportion of them have been assigned to a certain category, the new document is also assigned to this category, otherwise not . Additionally, finding the nearest neighbors can be quickened using traditional indexing methods. To decide whether a message is legitimate or not, we look at the class of the messages that are closest to it. The comparison between the vectors is a real time process. This is the idea of the k-nearest neighbor algorithm:

3.9.5 Decision Tree Classifier:

Decision tree classifier is a machine learning algorithm [4], which has been widely used since the last decade for classification is algorithm applies a simple method of solving any problem of classification. A decision tree classifier is a collection of well defined questions about test record attributes. Each time we get an answer, a follow up question is raised until a decision is not made on the record. Tree-based decision algorithms define models that are constructed iteratively or recurrently based on the data provided. The decision tree based algorithms goal is used to predict a target variable's value on a given set of input values. This algorithm uses a tree structure to solve classification and regression problems. Figure 3.7 shows the basic structure of the decision tree.

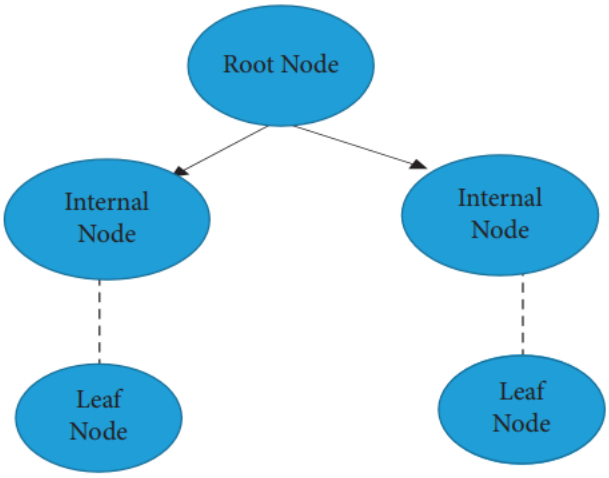

Figure-3.7: Decision tree

3.10 Performance metrics:

Performance metrics are used to evaluate the accuracy and effectiveness of machine learning models. In this section, we describe several commonly used performance metrics.

3.10.1 Accuracy:

In machine learning, accuracy refers to the degree to which a model can predict the correct outcome. This measure is determined by dividing the total number of correct predictions by the total number of predictions made by the model. The formula for accuracy is,

$$\text{accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{False Positives} + \text{True Negatives} + \text{False Negatives}}$$

Where, True Positive are the correctly predicted positive cases True Negatives are the correctly predicted negative cases, False Positives are the incorrectly predicted positive cases, and False Negative are the incorrectly predicted negative cases.

3.10.2 Precision:

Precision is a performance metric that evaluates the accuracy of positive predictions made by a model. In other words, it measures the proportion of true positives out of all predicted positives. To compute precision, we divide the number of true positive cases by the total number of positive predictions. A higher precision score indicates that the model is more precise and makes fewer false positive predictions. The formula for precision is:

$$\text{precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

3.10.3 Recall:

Recall is a statistical measure that evaluates a model's ability to identify all relevant cases in a dataset. In other words, recall measures how well the model can "recall" or correctly identify all positive cases in a dataset. It is calculated by dividing the number of true positive cases by the total number of actual positive cases. A high recall score indicates that the model is effective at identifying positive cases, while a low score suggests that the model is missing many positive cases. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3.10.4 F1-Score:

The F1-Score is a metric that determines the trade-off between precision and recall. It calculates the harmonic mean of precision and recall, with a range of values from 0 to 1. A higher F1-Score indicates a better balance between precision and recall, while a lower score indicates an imbalance between the two metrics. The formula for F1-Score is:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

3.11 Experimental Setup:

Python and its libraries are used for data collection, dataset construction, and model implementation and evaluation. The most popular and commonly used Python libraries for data manipulation were used in this project. The following is a list of the used libraries, their versions and what they are used for:

1. Natural Language Toolkit- Provides a set of diverse algorithms for NLP.
2. Regular expression - Specifies a set of strings that matches
3. Metplotlib – Data visualization.
4. NumPy - Data collection, dataset construction.
5. Pandas - Data collection, dataset construction.
6. Scikit-learn - Model evaluation, dataset splitting.

The computations for the model training and evaluation are done on the jupyter notebook.

3.12 Summary:

In brief, we have discussed several machine learning methodologies such as SVMs, Naive Bayes, KNN, Decision tree, LR along with the evaluation metrics accuracy, precision, recall, and F1-score. SVMs are useful for finding the optimal boundary between classes. Naive Bayes is a probability-based classifier used in document classification, and LR makes quick and accurate predictions. Evaluation metrics such as accuracy, precision, recall, and F1-score are important for measuring the performance of these models. Overall, machine learning provides effective algorithms for predictive modeling, and evaluation metrics help measure their performance. Understanding the strengths and weaknesses of each method and selecting the appropriate evaluation metric can help build accurate and reliable predictive models.

Chapter 4
Result and Discussion

4.1 Introduction:

In this section, we present the outcomes of our analysis and examine the implications of these discoveries for the creation of more precise and efficient spam detection model. We also discuss the difficulties we faced during the analysis and the techniques we employed to overcome them. Our discoveries offer useful insights into the application of machine learning algorithms for spam detection and may have significant consequences for the development of more effective strategies in the future.

4.2 Performance Analysis:

This study assessed the effectiveness of various machine learning classifiers for detecting spam using datasets. Their average performance scores were calculated.

For feature selection, 6677 features were extracted using TFIDF . These selected features were then used to train the model, and the SVM, LR, NB, DT, and KNN machine learning classifiers were employed to predict the performance. We have trained 80% data and test 20% data.

4.2.1 Model Selection:

We summarize the performance result of the five machine learning methods in term of precision and accuracy. Table-1 summarize the results of the five classifiers. In term of accuracy we can find that the SVC method is the most accurate while the Naïve Bayes, K-nearest Neighbor, Decision tree give us approximately the same lower percentage, while in term of spam precision we can find that the Naïve bayes & K-nearest Neighbor method has the highest precision among the algorithms while LR, SVC method have very competitive percent.

	Algorithm	Accuracy	Precision	Recall	F_Measure
1	KN	0.900387	1.000000	0.253623	0.404624
2	NB	0.959381	1.000000	0.695652	0.820513
0	SVC	0.972921	0.974138	0.818841	0.889764
4	LR	0.951644	0.940000	0.681159	0.789916
3	DT	0.935203	0.838095	0.637681	0.724280

Table 4.1 Performance of five machine learning algorithms

The decision tree has the worst precision percentage. The performance of the NB and KN have the same precision 1 (target value) but NB shows more accuracy than KN while the SVC provide comparatively moderate accuracy and precision and LR gives satisfying performance than DT. SVC gives highest recall value and F-measure value among five machine learning algorithms.

But overall NB shows best performance in overall characteristics so that is why we choose this algorithm. Therefore Naïve Bayes was selected to build the final classifier model.

In addition we have included a bar graph illustrating the performance of each algorithm in terms of accuracy, precision, recall & F-1 score.

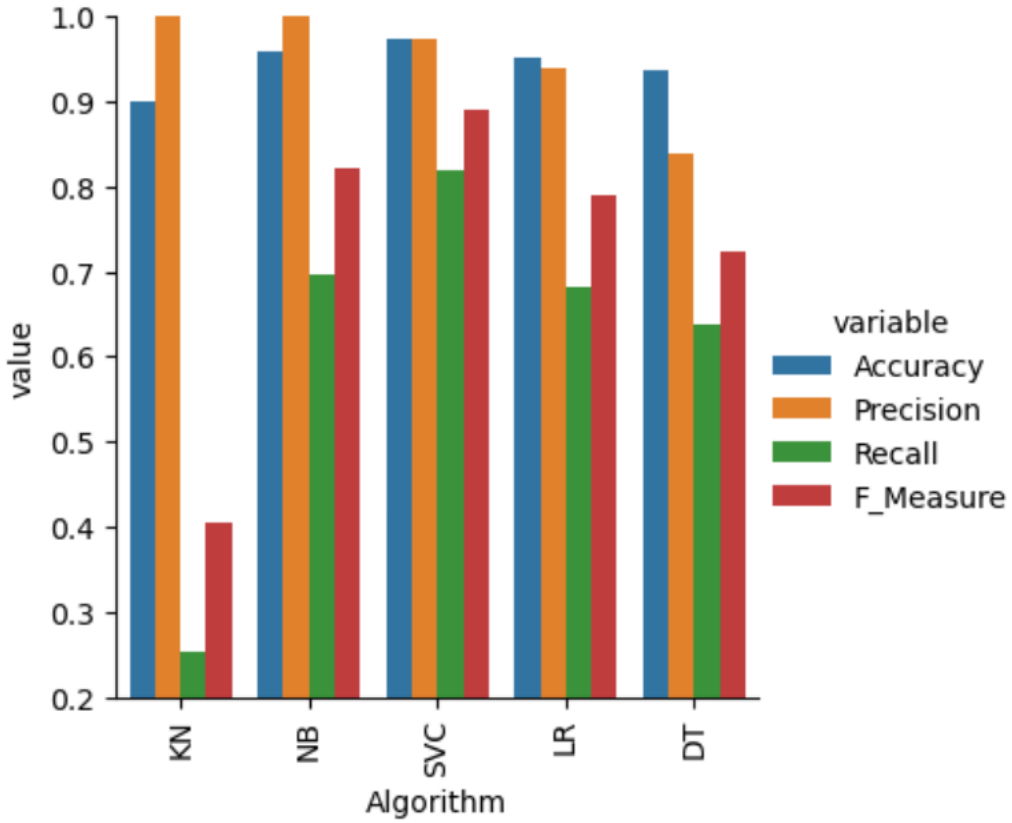

Figure 4.1 A performance bar graph of five machine learning algorithms.

4.2.2 Performance Analysis of Built Model:

After the selection of the best model NB, the final multiclass supervised classification model is ready to be trained and built. In this circumstance, the selected k-best features and the optimized hyper parameters were used to train the NB model. Afterwards, the model performed prediction on 6,677 words as testing dataset. Initially, as the size of feature changes, the accuracy of the model also changes gradually. Selecting k=2000, the model gained its maximum accuracy of 97.68% but thereafter the proportion continued to decline as the feature size changes. And using all the TFIDF features the model achieved an accuracy of 95.93%.

Number of words	Accuracy	Precision	Recall	F-Measure
2000	0.976789	1	0.826087	0.904762
2500	0.975822	1	0.818841	0.900398
3000	0.970986	1	0.782609	0.878049
3500	0.970019	1	0.775362	0.873469
4000	0.968085	1	0.760870	0.864198
4500	0.967118	1	0.753623	0.859504

Table 4.2 Performance evaluation

In addition we have included a graph illustrating the performance of NB algorithm in terms of accuracy, precision, recall & F-1 score.

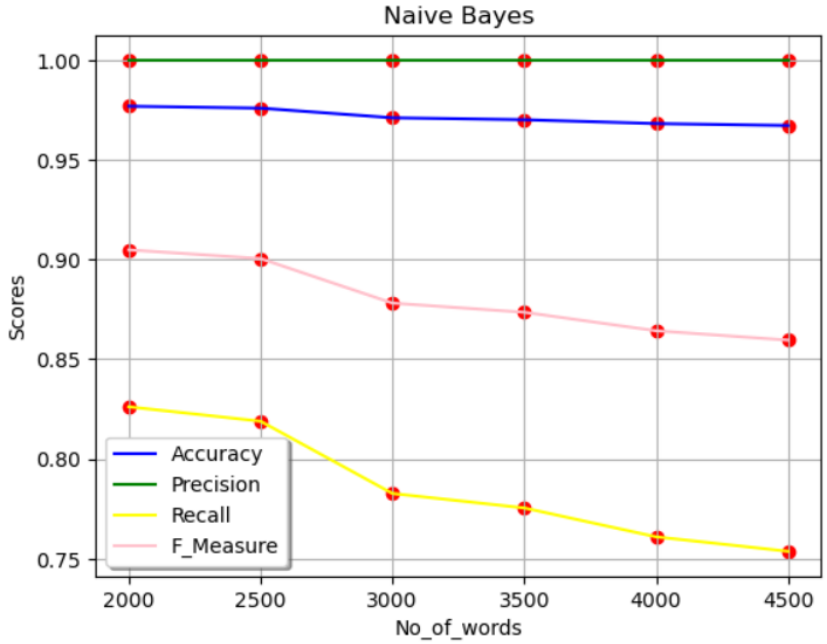

Figure 4.2 Performance evaluation

4.3 Challenges Faced and Solutions:

The study encountered several challenges such as model selection, text feature extracting but we successfully tackled them by analysing and optimizing machine learning algorithms, preprocessing data carefully, and utilizing computational resources efficiently. By overcoming these challenges, the study has provided significant insights into the use of machine learning algorithms for spam detection, which can potentially lead to the development of more accurate

and efficient detection strategies in the future. Overall, we have demonstrated how we faced the challenges and showcased the potential of machine learning algorithms in detecting depression, paving the way for further research in this area.

4.4 Research Gaps and Open Research Problems:

This section discusses the research gaps and open research problems of the spam detection and filtration domain. In the future, experiments and models should be trained on real-life data rather than manually created datasets, because, in the various article, the models trained on artificial datasets perform very poorly on real-life data. The following are some other future research directions and open research problems in the domain of spam detection.

i. Some studies considered header, subject of the email, and message body as a feature for spam classification. While these features are not enough for fully accurate results, manual feature selection and features should also be.
ii. Almost all researchers presented their results based on accuracy, precision, recall, etc., while the time complexity of machine learning models should be considered an evaluation metric.
iii. Some researchers show promising results in the process of feature extraction using a bag of words. They claim that the email header is as important for spam detection as the content of the body. So, deep feature extraction of the header line should be considered.
iv. Fault tolerance, self-learning, and quick response time can be better by using comprehensive feature engineering and an accurate preprocessing phase.
v. Deep learning models with dynamic updating of feature space are needed to implement for better spam classification. Most of the current filters cannot update their feature space.
vi. The security of spam detection and filtration system is needed for better accuracy and reliable results.

Chapter 5
Conclusion and Future Work

5.1 Conclusion:

Due to the increase in number of spam emails by the users, email spam has become one of the most demanding research topics. Various methods are used by different authors for spam email classification. We have used the concept of Naïve Bayes, Support Vector Machine, K-nearest Neighbor, Decision tree, LR for spam email detection. The evaluation of the experiment is done on the basis of fmeasure, precision, accuracy and recall. By evaluating the results, we can say that the accuracy and precision percentage among of the five methods and Naïve Bayes shows a very satisfying performance. The study provides comprehensive insights of these algorithms and some future research directions for email spam detection and filtering.

5.1 Future work:

In future work, we plan to extend our study to explore the use of larger and more diverse datasets to improve the performance and robustness of our models. Additionally, we aim to explore deep learning approach for optimizing the efficiency and scalability of spam detection of email, which would involve exploring several new approaches to make the models faster and more reliable.

References

[1] S. S. ,. N. Nikhil Kumar, "Email Spam Detection Using Machine Learning Algorithms," *Inventive Research in Computing Applications*, pp. 108-113, 2020.

[2] S. E. W.A. Awad, "Machine Learning Methods For Spam Email Classification," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 3, pp. 173-184, 2011.

[3] A. S. J. A. K. K. Nandan Parmar, "Email Spam Detection using Naïve Bayes and Particle Swarm Optimization," *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY* , vol. 6, no. 10, pp. 367-373, 2020.

[4] R. A. A. D. K. B. A. S. Naeem Ahmed, "Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges," *Security and Communication Networks*, vol. 2022, 2022.

[5] N. R.Deepa Lakshmi, "Supervised Learning Approach for Spam Classification Analysis using Data Mining Tools," *(IJCSE) International Journal on Computer Science and Engineering*, vol. 2, pp. 27602766 , 2010.

[6] T. Almeida, J. M. Gómez Hidalgo, and A. Yamakami, "Sms spam collection dataset." https://www.kaggle.com/uciml/sms-spam-collection-dataset, 2011. Accessed: February 17, 2024

[7] Renuka, Karthika D., and P. Visalakshi. "Latent semantic Indexing primarily based SVM Model for Email Spam Classification." (2014).

[8] Feng, Weimiao, Jianguo Sun, Liguo Zhang, Cuiling Cao, and Qing Yang. "A support vector machine primarily based naive Bayes algorithmic program for spam filtering." In Performance Computing and Communications Conference (IPCCC), 2016 IEEE thirty fifth International, pp. 1-8. IEEE, 2016.

[9] Kumaresan, T., and C. Palanisamy. "E-mail spam classification using S-cuckoo search and support vector machine." International Journal of Bio-Inspired Computation 9, no. 3 (2017): 142- 156.

[10] Olatunji, Sunday Olusanya. "Extreme Learning machines and Support Vector Machines models for email spam detection." In Electrical and Computer Engineering (CCECE), 2017 IEEE 30th Canadian Conference on, pp. 1-6. IEEE, 2017.

[11] M. S. M. E. A. M. N. H. K. R. Pankaj Bhowmik, "Analysis of Social Media Data to Classify and Detect Frequent Issues Using Machine Learning Approach," *ResearchGate*, pp. 393-398, 2020.