Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Email Spam Detection Using Content, Semantic, and Entropy-based Features

A thesis submitted to school of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of Master of Science in Telecommunication Engineering.

By

Tigist Beyene

Advisor

Dr. Surafel Lemma

September 2021

Addis Ababa, Ethiopia

Declaration

I, the undersigned, declare that the thesis comprises my original work in compliance with internationally accepted practices; I have fully acknowledged and referred all materials used in this thesis work.

Tigist Beyene

Name	Signature

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

This is to certify that the thesis prepared by Tigist Beyene, entitled *Email Spam Detection Using Content, Semantic, Entropy based Features* and submitted in partial fulfillment of the requirements for the degree of Master of Science Telecommunication Engineering complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Advisor Dr. Surafel Lemma Signature _____ Date _____

Evaluator(1) Dr. Yalemzewd Negash Signature _____ Date _____

Evaluator(2) Dr. Fitsum Assamnew Signature _____ Date _____

_____

Dean, School of Electrical and Computer Engineering

ABSTRACT

The spam detection technique helps the business prevent unnecessary emails from reaching inboxes and preventing any consequential harm to an organization or user. To train the machine learning algorithms, several researchers use features extracted from the email. These features, however, capture only the content and are computed per email message. However, information aggregated per sender, i.e., content, entropy, and spam email similarity per sender, are not studied. In this study, we propose to use additional features that capture the content, similarity, and entropy of emails sent by the same sender. In this regard, we extracted six new features that could help to improve spam email detection. The six features are number of emails, duration, character length of the sender address, number of recipients, the similarity between emails, and entropy value within the sender subject. To build the prediction model, we used four machine learning algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. The proposed approach is evaluated using a dataset collected from ethiotelecom. The results show that the dataset augmented with the new features improves email spam detection performance. The F1-score of email spam detection is improved by 6.6%, 9.9%, 20.7%, and 11.3% using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively. The overall improvement is 12.1% on average. Among the Four algorithms used to build the predictive models, Random Forest performs better in detecting spam emails. We computed feature importance using the Information Gain and Gain Ratio algorithms to see which features helped to improve email spam detection. The result shows that the new features, length of address, the number of receivers, duration, sent emails, and entropy, are in the top six ranks. This indicates that the newly introduced features contributed to the improvement seen in email spam detection.

KEYWORDS

Email Spam detection, Classification, Supervised Machine Learning, proposed new features, Feature importance.

.

ACKNOWLEDGMENTS

First, I thank God for his absolute protection and giving me the strength, patience, and knowledge to complete this research. I want to thank my husband, Ashenafi, for his continued support. He was always around. Along with him, I would like to thank my daughter Soliyana and my son Yohannes for waiting for my ignorance and patience during my thesis writing. On the other hand, my sincere regard goes to my father, mother, brother, sisters, and family here and abroad for their love and support.

Second, I would like to thank Dr. Surafel Lemma for his priceless advice, continuous follow-up, and guidance during this thesis. His observation, straightforward advice, and support were valuable and constructive.

I greatly appreciate and acknowledge AAiT collaborating with ethiotelecom for their devotion and sponsorship to make this postgraduate program fruitful. Finally, I would like to thank the ethiotelecom security division staff for providing me the data for the email spam detection study and input for the thesis.

TABLE OF CONTENTS

List of Figures

List of Tables

ACRONYMS

KNN	K Nearest Neighbor
SVM	Support Vector Machine
RF	Random Forest
LR	Logistic Regression
WEKA	Waikato Environment for Knowledge Analysis
FP	False Positive
FN	False Negative
TP	True Positive
TN	True Negative
AUC	Area Under the Curve
ROC	Receiver Operating Characteristics
FPR	False Positive Rate
ISP	Internet Service Provider
AI	Artificial Intelligence
CEO	Chief Executive Officer
SMTP	Simple Mail Transfer Protocol
RPC	Remote Procedure Call
ML	Machine Learning
IG	Information Gain
GR	Gain Ratio
NLP	Natural Language Processing
MMH	Maximum Marginal Hyperplane
DT	Decision Tree

INTRODUCTION

Telecom service providers use email services for their day-to-day operational activities. By using this application, they send and receive sensitive data. Hence, email security becomes more critical in any operator/organization. Email security refers to protecting email users from various attacks [9] - [11]. There are many types of email security threats such as Viruses, Phishing, Man-In-The-Middle, Eavesdropping, Dictionary attacks, Spam, and Denial of service attacks [9] - [11]. One of the mechanisms to secure email users is the use of proper email spam filtering techniques.

Spam emails are annoying to most users, and users receive emails without the users knowledge. The growing problem of email spam motivates the emergence of email spam detection and filtering technique. Email spam detection techniques can broadly be classified as rule-based and machine learning-based techniques. The rule-based techniques work by setting a set of rules for classification. Rules help to detect and filter incoming emails on the email server. However, the spammer's behavior is not statics, and their character change frequently. Hence, these techniques are not effective in filtering spams with new behavior. Rule-based techniques have more chances to increases the false positive and false negative values. The impact of false positives is that the technique considers essential emails as spam and filters them out. On the other hand, false negatives see spam emails as normal emails that will affect the user's activity. The impact of false-positive value to the user is that a particular email may be essential for the user because of the predicted result user may lose that email.

On the other hand, the user may be attacked by that spam email for the false-negative result since that email is spam. Many have suggested using machine learning approaches to detect and filter emails [11]. Machine learning techniques are the most widely used techniques for spam detection/filtering [11]. These techniques use data rather than predefined rules. The machine learning algorithms used for spam detection are Naïve Bayes, support vector machines, neural networks, K-nearest neighbor, rough sets, logistic regression, random forest [11]. These techniques use datasets with independent variables as input and predict the dependent variable as the output. In classification, the input dataset is split into two as training and testing. The training dataset is used for model building, while the testing dataset is used to evaluate the

model's performance. Machine learning techniques help improve email spam detection using different email features since the features are extracted from the behavior of the spam and non-spam data of that specific operator/organization.

Different researchers use different email features to improve email spam detection using machine learning algorithms. Depending on the selected features, the performance of the machine learning approaches also varies. For this research, we used twelve features; the following features are used in the state-of-the-art number of repetitive words, number of nouns, number of words, number of unique words, number of capitalized words, number of question marks. The remaining features like number of emails sent, duration, number of sender address characters, number of recipients, the similarity between emails, and entropy value(diversification of information) between emails are the newly added computed features.

1.1 Statement of the problem.

In general, email spam is a common problem for electronic email users. Spam mail may contain malicious codes that affect the mail system. Spam email may contain a virus that may create a problem for the entire system [1]. Without the request of the email user, spammers send bulks spam emails to different users [3]. Spammers are responsible for causing additional damages to various organizations. The damages include loss of revenue and employee productivity. Spammers may also send viruses using spam email. This type of attack affects the performance of an organization's network. To minimize the damage caused by these spam emails, different researchers conducted different researches on email spam detection techniques. Several researchers proposed rule-based and machine learning-based approaches to detect spam emails and filter them to address this problem. Rule-based systems use predefined rules to detect and filter spam emails. This approach, however, is not usually practical mainly because spammers change the characteristics of spam.

Machine learning approaches use datasets that contain features extracted from spam emails to build spam detection models. The features used in these approaches are content-based, entropy-based, and semantic-based features. However, these features do not cover essential aspects

(e.g., the similarity between emails within the same sender, diversification of information within the same sender) of spam emails. It lacks a holistic view. This research work predicting email as spam or non-spam by examining the data found. Despite these different research works, to the best of our knowledge, no research examines how to improve email spam detection per sender per six hours using the number of emails sent, duration, length of the email sender address, the similarity score, and entropy value. This research empirically examines the impact of the newly computed features per sender on email spam detection by considering the above issues.

To meet our aim, we conduct empirical experiments after augmenting the newly added computed features per sender with features in state-of-the-art found in [6] per message. Therefore, in the end, this research work tries to answer the following research questions.

RQ1: ***[Performance of augmented features]*** To what extent do the newly added computed features improve the performance of email spam detection?

RQ2: ***[Contribution of newly added features]*** Which of the newly added computed features contribute more to improving email spam detection?

1.2 Objective

1.2.1 General Objective

This research aims to improve the performance of machine learning-based email spam detection approaches by introducing newly computed features per sender.

1.2.2 Specific objective

The specific objective of this research are:-

- Study the characteristics of the collected spam data and add additional features.

- To evaluate the impacts of the newly added features on email spam detection improvement.
- To examine the importance of each feature in email spam detection.
- To compare the performance of the four commonly used machine learning algorithms from the email spam detection technique and choose the best performing algorithm.

1.3 Scope and Limitation

1.3.1 Scope of the thesis

This thesis work is limited to improving email spam detection using the newly added computed features presented in Section 3.1. This study uses email headers using machine learning techniques to improve email spam detection by considering adding new features. In general, the scope of this thesis can be summarized by the following significant points:

- To detect email spam, different researchers suggested a different approach. Those approaches include rule-based, machine-learning-based. All approaches have their advantage and disadvantage. In this research, we will focus on a machine learning-based approach.
- There are two ways to improve email spam detection in machine learning: one defines a new algorithm using the existing features, and the other identifies new features. In this research, we identify new features.
- We do not look for new algorithms. We used the commonly used algorithms.
- Prepare a dataset for the newly added features and augment it with the dataset containing state-of-the-art features.
- Examine whether the newly added features have an impact on the performance of email spam detection.

- Examine the importance of the newly added features over the state-of-the-art features.
- Prepare a dataset for the newly added features and augment it with the dataset containing state-of-the-art features.
- Examine whether the newly added features have an impact on the performance of email spam detection.
- Examine the importance of the newly added features over the state-of-the-art features.

1.3.2 Limitation of the thesis

The newly added features could be computed from the email's subject and the email's body. However, we are not able to get the body of the email. Our new features are computed from the subject of the email. We believe that subjects are representatives of the body.

1.4 Methodology

The methodology followed to achieve the general and specific objective of this thesis is:

1. Related work of literature is reviewed to understand the available email spam detection techniques. To examine the previous research works, an exhaustive survey is conducted and have future research directions by looking at the methodology used, algorithm applied, the performance metrics used, and the gap of previous research works.

2. Ethiotelecom mail spam detection architecture and email flow techniques are studied.

3. An informal meeting was conducted with domain experts regarding the common problem of spam email and the limitation of the existing email spam detection technique.

4. The datasets were collected from the ethiotelecom mailbox server and spam guarantee log.

5. Python programming language for preprocessing and WEKA(Waikato Environment for Knowledge Analysis) tool for the model building (training and testing) was chosen to ana

lyze the collected data since it is user-friendly and has plenty of algorithms, feature selection algorithms, and different analytical graphs.

6. To evaluate the performance of our model and the impact of our proposed features, we used commonly used evaluation metrics., Accuracy, FPR(False Positive Rate), precision, recall, F-measure, ROC &AUC(Receiver Operating Characteristics, Area Under the Curve) , derived from a standard confusion matrix.

1.5 Contribution of the research

Various researches were done on email spam detection using machine learning algorithms; Previously done research works are enormous in terms of quantity and the subject matter they try to address. However, none of them look at to compute the content, entropy, similarity-based features per sender and assess their impact on email spam detection techniques to the best of our knowledge. By taking the fact mentioned above into account, this research work targeted improving email spam detection by using the data found in ethiotelecom. Therefore, the main contributions of this research work are as follows:

- We introduced six new features to help improve the performance of machine learning-based email spam detection. Those new features are the number of recipients, the number of emails sent, the time interval of email sent, number of characters in the sender address, number of characters in the sender address, the similarity between email subject within the sender, entropy value between email subject within sender.
- We assessed the impact of sender-based features on the performance of email spam detection.
- We assessed the contribution of the newly added features in terms of:-
    - ✓ Performance in email spam detection.
    - ✓ Ranking in terms of contribution.

1.6 Thesis Organization

The thesis consists of six chapters. The First Chapter consists of the introduction, objective, scope, methodology, the contribution of the research, and thesis organization. Chapter Two presents the details of mail, mail flow, email spam, machine learning-based email spam detection, the benefit of email spam detection, email spam filtering architecture, and related works. Chapter Three describes the characteristics of spam email, the proposed feature, and methodology we follow to implement the new features for better machine learning-based email spam detection. The fourth chapter discusses the experiment setup, while the results are presented in Chapter Five. The Sixth chapter concludes the thesis and indicates future research directions.

2

LITERATUR REVIEW

2.1 Email

Email is an electronic means of communication for users that access the internet. The message is exchanged from one user to another through electronic means. It allows users to send and receive a message from one device to another. Due to the growth of internet users, email users have increased exponentially [4]. Nowadays, email is used for business, study, and different activities for handling their day-to-day operational activities. Its fast and cost-effective means of communication. This medium of communication has an additional flow when sending and receiving a message within the uses. Mail flow steps for sending and receiving email helps to see the coordination between different services within the various email servers.

2.2 Email Flow

This section mainly discussed how mail flow from internal and external senders enters the transport pipeline. Email flow views the steps for sending and receiving a message. In general, a mail sent to a user first comes to the edge transport server of the mail server. After the email message is processed at the server, it will be routed to the mailbox servers. Depending on the edge server, the way message from the external user enter the transport pipeline varies. The transport pipeline consists of a front-end transport service on mailbox servers; this service does not check message content, does not communicate with the mailbox transport service, and any message is not stored locally. Transport service on mailbox servers also performs message grouping and message checking; however, it never communicates directly with the mailbox database, mailbox transport service on mailbox servers, transport service on edge transport servers [15]. The mail flow could be classified as inbound and outbound mail flow. Below is a short description of these categories.

2.2.1 Inbound mail flow

The below figure shows the email from the external sender to the transport pipeline. A message from the external organization enters the transport pipeline through the edge transport server [14]. In the mailbox server, the front-end transport service accepts the message. After that, the message is transferred to the mailbox server transport service. The mailbox transport service uses its protocol to deliver the email to the local mailbox database [14].

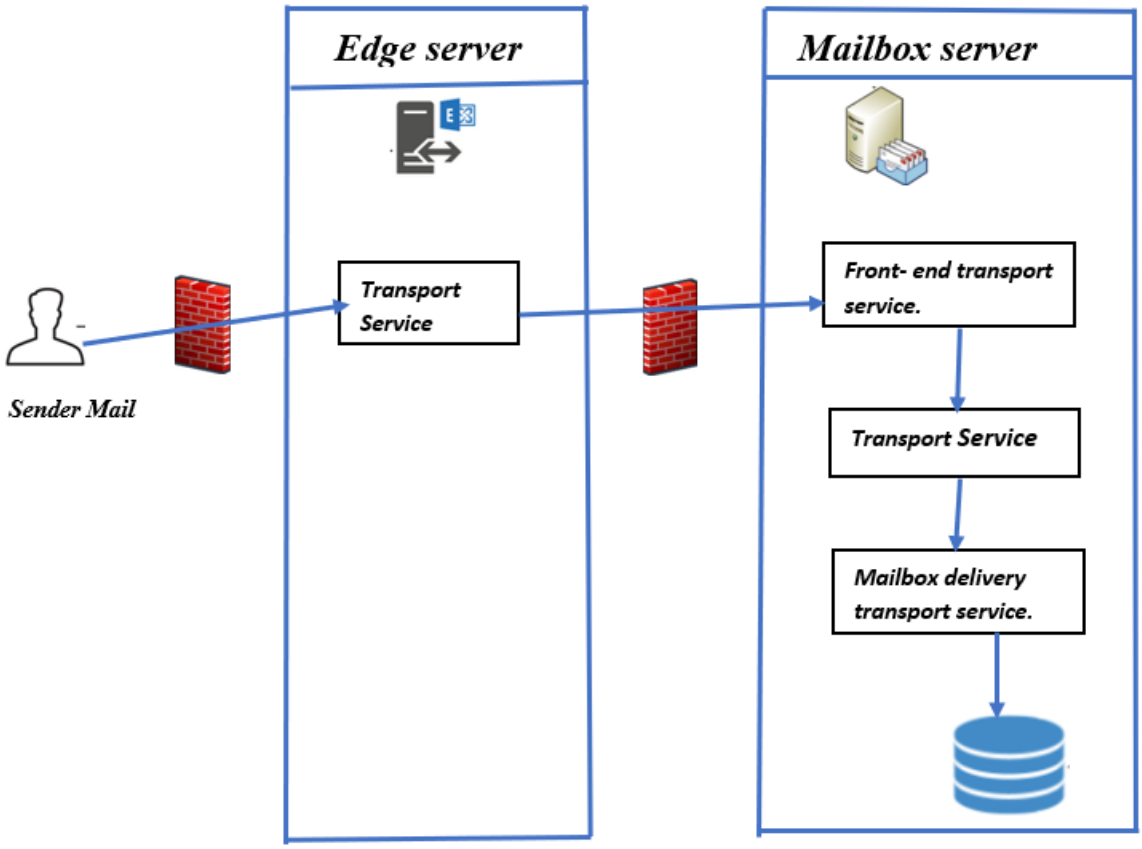


Figure 2.1 Inbound Mail flow[14].

2.2.2 Outbound mail flow

Figure 2.3 shows us the outbound mail flow**.** **T**he mailbox transport submission service under the mail server uses remote procedure call (RPC) protocol to retrieve the outbound message from the local mailbox database. Mailbox transport submission service uses a simple mail transfer protocol (SMTP) protocol to send the message to the transport service, and then it goes to the edge server.

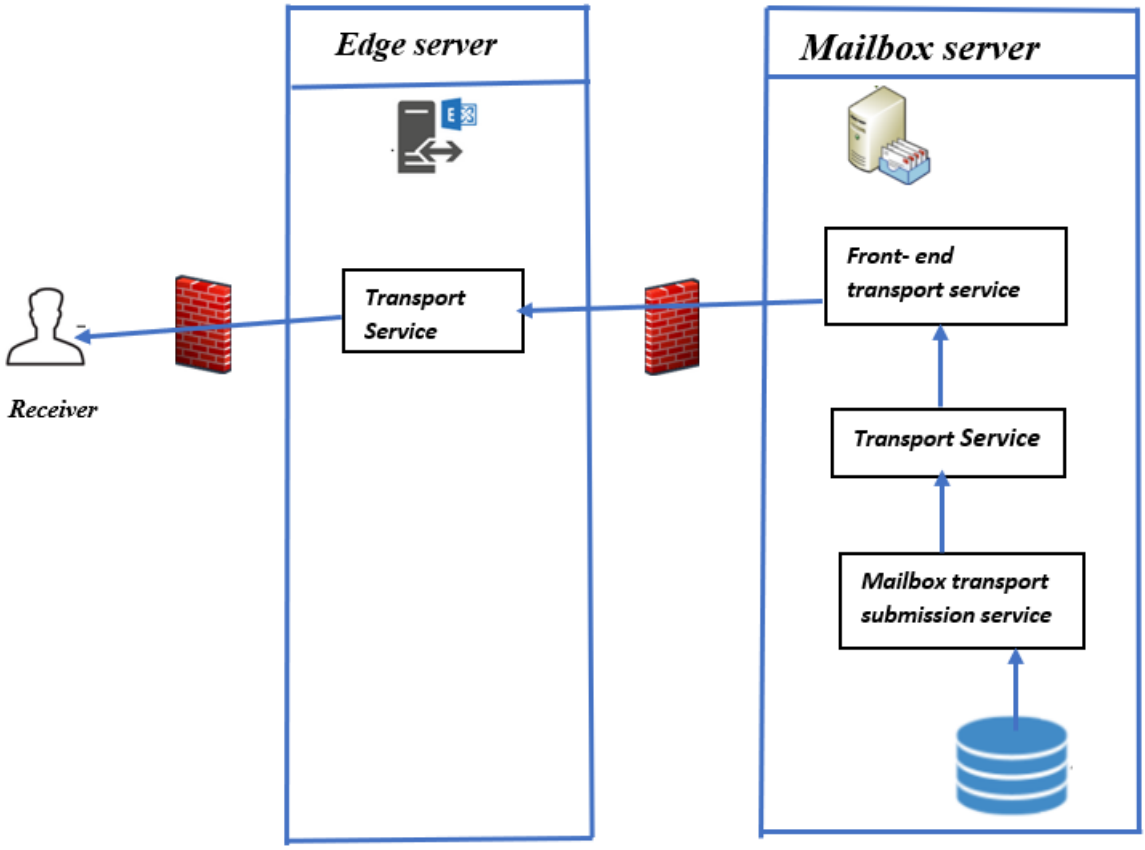


Figure 2.2 Outbound Mail flow[14].

2.3 Email Spam

Spam is an unsolicited bulk email that the user receives without the user's request [5][7]. Spammer's primary goal is to spread unnecessary electronic information to cause psychological and monetary harm to the victims. Spammers achieve the email servers by sending a mass quantity of unsolicited emails to end-users. Spam fills inbox with several ridiculous emails that may cause several damages to the system. Spam wastes bandwidth, brings harmful information to the user, causes a system crash, and decreases employee productivity.

2.4 Benefit of Email Spam Detection

Nowadays, a business's email account is riddled with unsolicited emails. For securing businesses and protect email users, spam detection is required. Detecting spam emails before they reach the end-user and, in general, the organizational network has several advantages. Below, we list some of the benefits of spam detection.

Protect from virus attack

Most of the time, spammers use spam email to cause some attack on the user application, network, and system. Through those spam emails, they can cause any threat. Most viruses, trojan horses, and worms are activated when we open an attachment or click a link contained in an email message. Spammers send a message with viruses. Therefore, detecting those emails will minimize the chance of attack by viruses.

Protect the company data

Through these spam emails, hackers can access the company data and use those data to cause damage to the company. Emails the recipient believes come from a legitimate source that requests the user to view a link or an attachment or go to a website and ask the user to enter personal information. Then the spammers access the user's data using the stolen data. Spammers send an email message and ask to deposit into their account, enter users personal

home or company network by getting the recipient to open an email attachment [28]. Therefore, email spam detection techniques help the user from such type of data loss.

Increase employee productivity

Spam emails, if not filtered, will appear in the inbox. These destruct and consume a relatively significant amount of users' time. Hence, users usually need to open almost all emails and check them to see if they are legitimate emails or spam. If the spam emails are too many, it is difficult for users to identify legitimate emails in time and respond. Spam detection and filtering approaches would help users minimize the burden of filtering legitimate emails and save their time.

2.5 Machine Learning based Email Spam Detection

Currently, email is the most widely used means of communication for the organization's day-to-day operational activity. Most organizations use machine learning techniques for making things simple. In this section, before we explain the machine learning algorithms which is used for email spam detection, we explain what machine learning is, and classification techniques will be explained in detail. Finally, classification algorithms will be described**.**

2.5.1 Machine learning

Machine learning is a field of study that gives computers the ability to learn and act like a human by inputting data without being programmed. It is a subset of artificial intelligence. Humans learn from experience machines do not have that "experience," so computers learn from data. Machine learning makes decisions and predictions based on past data. In the machine learning approach, there are supervised learning, unsupervised learning, semi-supervised, and reinforcement learning approaches. Below is a short description of these approaches which need data for learning.

Supervised learning

It is a machine learning approach that uses a labeled data for prediction. Supervised learning can further be classified as *classification and regression*. The main goal is to predict the outcome of the new data.

- **Classification** is a process of predicting class or category from the actual value. Classification problems use an algorithm to assign test data to the correct type correctly. To implement classification, we primarily train the classifier then it will be used for prediction.
- **Regression** is another type of supervised learning approach it helps to learn the relationship between dependent and independent attributes.

Unsupervised learning

It's another machine learning approach that uses machine learning algorithms to cluster and analyze the unlabeled dataset. It's used for three main tasks *clustering, association, and dimensionality reduction*:

- **Clustering** is a data mining technique based on the similarity and difference group unlabeled data.
- **Association** is another type of unsupervised learning method for finding the relationship between variables. It uses different rules.
- **Dimensionality reduction** is another unsupervised learning method used when the number of features in each dataset is too high.

Semi-Supervised learning

It is a type of machine learning algorithm that is between supervised and unsupervised learning. It contains both labeled and unlabeled datasets. This technique allows learning the algorithm from a small amount of labeled data and a vast amount of unlabeled data for predicting the output, which helps to benefit from supervised and unsupervised techniques.

2.5.2 Classification in machine learning

Classification is the process of categorizing a given set of data into classes. The class is either labeled, class, or categories. Classification is a supervised machine learning technique. It helps to predict the new data based on the training dataset. In our study, this technique is used.

2.5.3 Machine Learning algorithms for classification

The classification algorithms that researchers used from the literature review are listed here:- Decision Tree, Logistic Regression, Artificial Neural Network, K Nearest Neighbor, Support Vector Machine, Naive Bayes, Random Forest. Below is a short description of those algorithms.

Logistic Regression

It is a classification algorithm in machine learning that uses one or more independent variables to determine an outcome. independent variables are the input feature that we used for building and testing the predictive model. So, it will have two possible results only.

Naive Bayes

It is a classification algorithm based on naive Bayes' theorem, which assumes independence among predictors [2]. This algorithm takes that the presence of one feature in a class is unrelated to the other feature. We can take advantage of this feature of the Naïve Bayes algorithm to calculate the possible occurrence of words in an email and classify it as spam or non-spam [2]. This assumes that the values of a specific feature are independent of all the other features given in that class [7].

K Nearest Neighbor

It is a lazy learning algorithm that stores all instances corresponding to training data in n-dimensional space [7]. The prons of this algorithm are that with a small dataset, accuracy is high. It does not focus on constructing a general internal model. Instead, it works on storing an instance of data [7].

Decision Tree

It is a classification algorithm. Building the decision tree for the classification problem needs to calculate Information Gain and entropy for the features. This algorithm helps by creating a classification model in the form of a tree.

Random Forest

This is also a supervised machine learning algorithm. It is an ensemble learning method of classification. It operates by constructing multiple decision trees. It uses voting for the final prediction to improve the accuracy.

Artificial Neural Network

A neural network consists of arranged neurons in layers; they take some input vector and convert it into an output. The process involves each neuron taking input and applying a function that is often non-linear and then passes to the production to the next layer.

Support Vector Machine

SVM tries to find a hyperplane that separates two data classes in data space while maximizing their margin [14]. A classifier represents the training data as points separated into categories by a gap as wide as possible. New issues are then added to space by predicting which type they fall into and which area they will belong to.

2.6 Spam Detection Technique Implementation

This section explains the implementation of the email spam detection technique in the email server. Email server architecture consists of two parts edge server and the mailbox server. The edge server handles all the external incoming and outgoing mail exchanges; this server includes anti-spam and mail flow rules to protect the mail user from different threats, including viruses and spam emails [15]. If the organization has an edge server, anti-spam and antimalware are implemented in the edge server; if not, it will be implemented in the mailbox server; spammers use a different technique to send spam emails to the users. Using the anti-spam method on the gateway and the premises will be more efficacious [15]. The edge server has different anti-spam agents (connection filtering agent, sender filter agent, recipient filter agent, sender ID agent, content filter agent, protocol analysis agent (sender reputation), attachment filtering agent. The goal of spam detection is that to minimize the volume of unsolicited emails. In general, it manages the inbound emails, detects spam emails, and avoids any emails containing any threats or viruses. Mail user agents are email clients that help the user compose and read emails such as Microsoft outlook. Some filtering techniques are implemented in both the client and server [11]. Many ISPs implement spam detection tools in the edge or mailbox servers [15].

2.7 Related Works

Several approaches are proposed to detect email spam. Based on the approach used for email spam detection, the approaches could be broadly classified as rule-based and machine learning-based approaches.

2.7.1 Rule-based Approach

Rule-based email spam detection is one mechanism for email spam detection. By Setting some rules, if the incoming email fulfills that rule, that email is predicted as spam or non-spam. The approach is mainly applied to the client-side and server-side filtering techniques. A client can receive and send emails through ISP. a client can filter spam emails by installing some framework to their PC by specifying some threshold [3]. An enterprise-level spam filtering framework is installed on the mail server [3].

The main drawbacks of the rule-based spam filtering technique are if the incoming spam email does not fulfill the threshold they set, that spam email may be considered as non-spam and enter the user inbox. This, and hence, increases the false negatives. If the threshold value is set to a higher value, the approach will consider standard emails as spam, increasing the false positives. Another drawback of this approach is that the behavior of spam emails changed dynamically, so, at this condition, this approach considers spam emails as non-spam.

2.7.2 Machine Learning-Based Approach

Sultana et al.[1] used a machine learning approach to detect spam emails and identify the IP address of the spammer. The approach block all IP addresses of the spammer. Blocking the spammer IP, however, may cost more when there is a false negative. The proposed system uses a naïve Bayes algorithm. The author uses the Kaggle dataset for training and testing the algorithm. However, the researcher does not use the sender information or other computed features to improve the email spam detection technique. Nandhini, S et al.[2] conducts another literature that followed such an approach. Nandhini. S et al. compare the five machine learning

algorithms using the UCI dataset. Using different performance matrices, they try to evaluate each algorithm, however for email spam detection, comparing the algorithm is not that useful for improving the performance of email spam detection; instead, identify which algorithm is better. Hanif Bhuiyan et al.[3] also explain several techniques such as standard email spam filtering technique, client-side and enterprise filtering spam detection, case-based email spam filtering. Under this, they explain content filter, header filter, blacklist filter, rule-based filter, challenge filter, permission filter, and challenge-response filter. The content filter used the email content and machine learning technique for detecting email as spam or non-spam. The other filtering technique uses the header information. Overall, this filtering technique help to improve email spam detection. Client and enterprise-based email spam filtering techniques based on a framework help secure the email spam transaction between two points. The user can filter the spam by installing different filtering techniques on their PC.this framework has integration with the mail user agent(MUA) and filters spam on the client machine. The enterprise-level spam filtering framework is installed in the mail server, integrating with a message transfer agent(MTA) to classify the email as spam or non-spam. This method helps to quickly detect the email spam detection false positive and false negative results improves email spam detection. The third email spam filtering method is the Case-Base spam filtering technique [2]. It is a famous method of machine learning technique. First, it collects the spam and non-spam data, then preprocessing those data, then classifications and analyses the techniques. In this approach, the final decision for classifying email as spam or non-spam is decided using self-observation and classifier results.

In Shubhangi Suryawanshi, et al.[4] the authors evaluate and analyze different machine learning approaches using the Kaggle and UCI Dataset; they use various machine learning algorithms to assess each model's performance. Most of the time, using different performance metrics helps evaluate the model from a different angle and see the accuracy. They aim to select the best algorithm.

Ammara Zamir, et al.[6] uses feature centric concept, which means the authors use different feature categories like content-based, entropy-based, semantic-based, user-based, lexical-based features. In this research, it is seen that the more the features in the dataset, the better the accuracy. Regarding the content-based features, using different measurements may capture the spammer's characteristics from the content information. The similarity score between the title and the content also helps to know the spammer's behavior since mostly spammer sends different titles and content for most users. The researchers build a different model for different feature categories that help to improve spam detection with computed features. The computed feature is extracted from understanding the behavior of spam emails. So, these features quickly characterized the behavior of the spammer and identified the spammer. The limitation of this approach is that they consider feature value within-subject and content of the email for some features per each email. Still, they can see between email subjects and between email contents within the same sender.

Similar to Ammara Zamir, et al.[6], Min Zhou et al.[14] a feature-centric approach for improving email spam detection. The researchers consider entropy-based spam detection. Using the entropy value of the subject and the content of the email to see if emails sent are diversified. The authors conducted two experiments to evaluate the impact of entropy-based features using two datasets. The results show that the entropy-based features helped to decrease false positives and false negatives.

According to Vinodhini. M et al. [12], highly motivated spammers cheat and destroy company data using spam emails because of the high usage of emails. Identifying this spammer and spam emails is a crucial subject of study nowadays in a vast parallel number of experiments conducted. In their study, they used machine learning-based algorithms. They used four machine learning-based features to gather reliable data and improve email spam detection: user-behavioral, user-linguistic, review-behavioral, and review-linguistic. The authors proposed the random forest classification algorithms with the NLP concept using the Twitter dataset using the above-listed four features and eight NLP concepts. In this paper, the authors use machine learning algorithms to identify the spammer and spam messages from the Twitter dataset. In addition to this, they also identify spammer's way of writing, which also helps to identify other spammers. They consider two feature sets which consist of content and user's behavior. It is

determined with the help of average content similitude, maximum content similitude, a ratio of exclamation sentences, and the ratio of first personal pronouns. The user behavior is determined with the help of properties such as reviews written, and an average negative ratio given. However, the limitation that we observe in this study is that they try to detect spam emails and spammers from user's behavior. However, in our research, we will add sender information for better performance.

3

PROPOSED APPROACH

This chapter presents and discusses the characteristic of spam and non-spam data and then the computed features we added based on spam and non-spam data characteristics. Finally, the methodology that we follow for implementing those newly added features using the machine learning technique.

3.1 Characteristics of Spam Emails

Nowadays, in any telecom operator and organization, there is plenty of spam and non-spam email data. The characteristic of these data is helpful to improve email spam detection. Table 3.1 summarizes the essential characteristics of spam email from these email spam characteristics; using those characteristics, we added additional new features.

Table 3.1 The characteristic of spam emails

No	Characteristic of spam email data	Description
1.	Spam emails are sent out any time of the day.	There is no specific time of the day at which spam emails are sent out.
2.	The sender address of a spam email is usually long.	The character length of the sender address for the spammer is more extensive than the legitimate users. Counting the number of characters on the sender address can help us differentiate the spam from non-spam email.
3.	Spam emails usually have similar subjects	Spam emails are usually composed once and sent out multiple times to different recipients without almost no change to both the subject and body.
4.	Spam emails are sent repeatedly to a group of recipients.	Spammers sent excessive emails to specific receivers in a given period. Legitimate users, however, may send more emails for different recipients at different times, not for a group of receivers.
5.	The message body of spam emails is usually similar.	From the characteristic of spam emails, spammers sent the same message body content to the same recipients. However, legitimate users send different message body content to the same recipients.

3.2 Proposed Features

To enhance the detection of email spam, we propose to augment the existing features with content-based, semantic, and entropy-based features. Content-based features aim to capture different aspects of spam email content. On the other hand, the semantics-based feature aims to capture the semantics using similarity measures between subjects of different emails. In contrast, the entropy-based feature captures the diversity of the subject of the email. Table 3.2 presents the summary of the proposed features derived from the characteristic of spam emails from Table 3.1. In the following sub-sections, we describe each feature along with a justification for selecting them. The features are computed per six hours. We chose this specific period because it needs some fixed intervals since we computed the feature by aggregating the sender information. After this short interval, our system predicts the email as spam or non-spam.

Table 3.2 Newly added Computed Features.

Feature type	Description	Metrics
Content-Based	Number of recipients	Count of the recipient per sender per six hours.
	Number of emails sent	Count of emails sent per sender per six hours.
	Time interval of email sent	Mean of time interval per sender per six hours.
	Number of characters in the sender address	Count of characters in the sender address.
Semantic Based	The similarity between email subjects within sender.	The similarity score between email subjects per sender per six hours per receiver.
Entropy-Based	Entropy value between email subject within sender.	Entropy value between email subjects per sender per six hours per receiver.

Below is a short description of each feature as per the category content-based, similarity-based, entropy-based. And the reason for selecting this feature is in parallel with the characteristic of the spam email.

3.2.1 Content-Based

From any organization, email data perspective, content-based features improve email spam detection [6]. It means that using the content of the email its possible to compute relevant features. For this research, using the subject content of the user email, the listed features are computed: the number of recipients of the email, the number of emails sent, an interval of email sent, character length of the sender address, all with a specific period. Below is a short description of these features.

Number of recipients

We added this feature considering as an expert by considering the following characteristics of spam and non-spam emails. That is, one spammer may send extensive emails to one user. However, when we see the non-spam users, extensive emails are not sent to one user but rather to different recipients. So, knowing this feature helps to improve email spam detection performance by counting the recipient per sender-specific period.

Number of emails sent

Counting the number of emails sent per sender per recipient will also help improve the performance of email spam detection. We added this feature as an expert by considering the following characteristics of spam and non-spam emails. One recipient may receive too many emails from the same sender. However, in non-spam emails, one recipient may receive few emails from the same sender per specific period, so knowing this helps improve email spam detection. This feature is computed by counting the number of emails sent per sender per recipient within a particular period. That is, spammers send many emails per day other than the non-spam users.

Time interval of email sent

Spammers usually repeatedly send several emails in short time intervals. However, legitimate users take a relatively long time. These characteristics, hence, could help distinguish spam email from a non-spam email. This feature is computed by taking the interval between emails for a specific recipient per sender within a particular period. Since the sender aggregates data, the final duration is considered by taking the average of the interval within the recipient per sender.

Number of characters in the sender address

The sender address of spam emails is usually longer than legitimate emails. Hence, it could serve as a good feature that can differentiate spam emails from legitimate emails. This feature is computed by counting the number of characters in the sender's address.

3.2.2 Semantic-Based

Semantic similarity helps to check the similarity between two documents. For our study, it checks the content similarity between email subjects of the same recipients. From the characteristic of the spam and non-spam data, Spammers sent the same subject content to the same recipient per specific period. However, legitimate users sent different subject content to the same recipient per particular period. Knowing this similarity score helps us identify spam from non-spam and improve the performance of email spam detection. Since the sender aggregates the data, consider the average similarity score. The cosine similarity function is used to compute the similarity score between subjects. If the subject content between emails is not similar, the similarity score will be round zero; if it is identical, the score will be higher. Below is the feature that we compute for knowing the similarity score between subjects.

The similarity score between email subjects by the sender

The benefit of this feature is discussed above. This feature is computed within the subjects. Its value evaluates the similarity between subject contents per recipient. For example, let X is the sequence of email subjects within the same sender, i.e., X = X1, X2,..., Xn. The number times the similarity scores are calculated using equation 3.3. The similarity score between two subjects is calculated using equation 3.2. Then we consider the mean similarity score value for the recipients.

Below is the general formula for the cosine similarity function.

$$\mathrm{Cos}\,\theta = \frac{A.B}{||A||.||B||} \tag{3.1}$$

Where:- A and B are the vector of subjects between the same recipients.

$$Cos\,\theta = \frac{\sum_{i=1}^{n} X1i\,X2i}{\sqrt{\sum_{i=1}^{n} X1i^2}\sqrt{\sum_{i=1}^{n} X2i^2}} \tag{3.2}$$

Where:- X1 and X2 are the subject of the email from the same recipient.

For Example, let say if under one recipient if there is Y number of subjects(emails). We calculate the cosine similarity score N times.

$$N = \sum_{i=1}^{n-1} n\text{-}i \tag{3.3}$$

Where:

n is the number of subjects

N is the number of times the similarity score calculated

3.2.3 Entropy-Based

Entropy measures how diverse a value is in a given content. It is a concept mainly used in information theory. When content is diverse, the entropy value is usually higher. Information content of non-spam emails is usually much more variable than spam emails [14]. Hence, entropy could be one of the discriminating features that would help to differentiate spam emails from standard emails. Below is a short description of this entropy-based feature.

Entropy value between email subjects within the same sender

We calculated the entropy value of the email subject within the sender. Since the data is aggregated using sender after calculating the entropy value between each, we consider the mean. For example, let X is the sequence of email subjects within the same sender, i.e., X = X1, X2,..., Xn. The number of times the entropy value is calculated by using equation 3.3. The entropy between two subjects is calculated by using equation 3.4. Then we consider the mean entropy value for the recipients. For Example, let say if under one recipient if there is Y number of subjects(emails). We calculate the entropy value score N times using equation 3.3

$$H(T) = \sum_{i=1}^{n} P(Ti)\ \log P(Ti) \qquad (3.4)$$

Where:

P(Ti) ----is the probability of each word within that subjects

3.3 Methodology

In this sub-section, we present the methodology followed to detect spam emails. Figure 3.1 shows the general structure of the methodology. To assess the impact of the newly added features, we used the approach followed in other similar studies [6] [4]. The main steps of the methodology are data collection, preprocessing, model building, and evaluation. Below we describe each step of the approach.

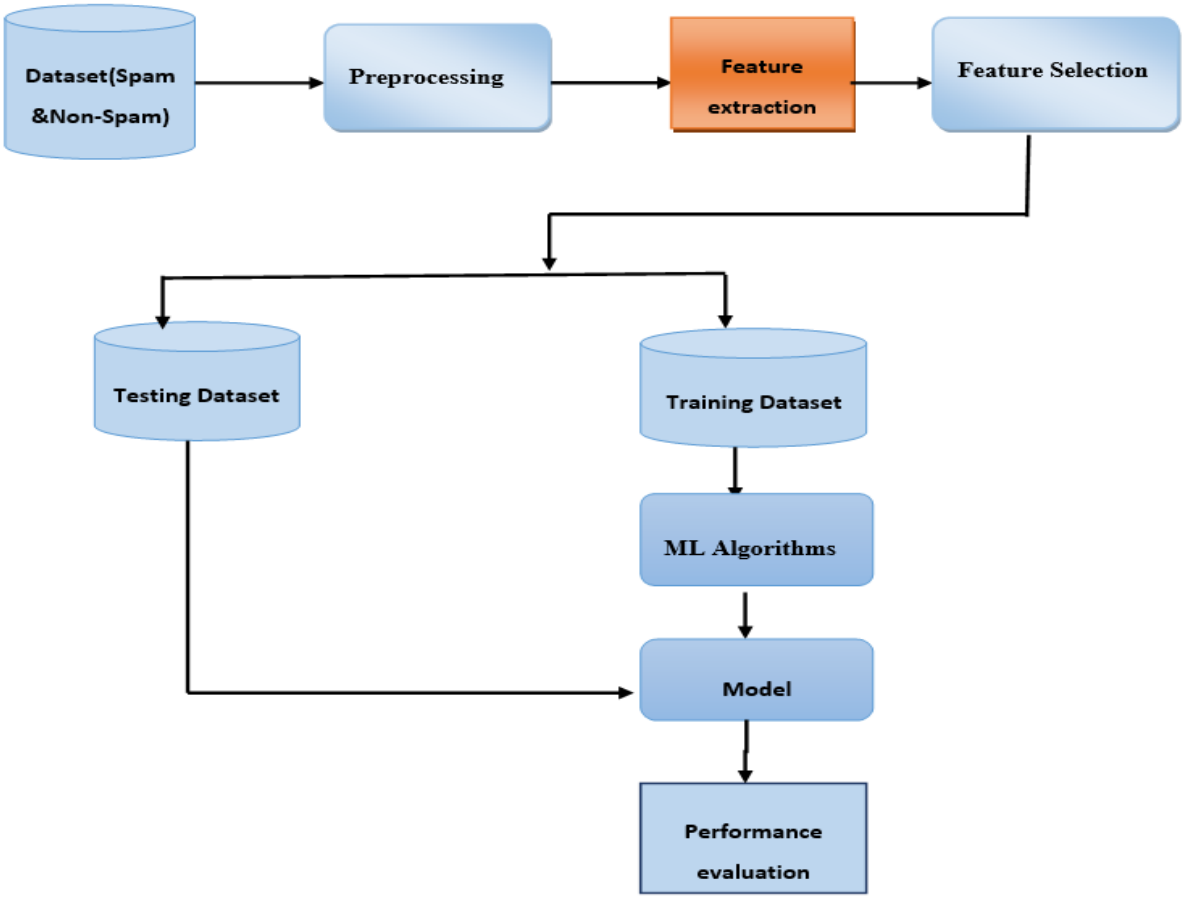


Figure 3. 1 Methodology

3.3.1 Data Collection

The base for email spam detection is the collection of spam and non-spam data. The data is aggregated by sender per six hours. It also collects the existing features that previous researcher used in [6].

3.3.2 Preprocessing

Data preprocessing is crucial for accuracy. The spam and non-spam data could be incomplete, and it needs some aggregation; hence the data must go through the preprocessing step. In this step, redundant and error data are removed from the dataset. Preprocessing includes tokenization, lemmatization, and stop word removal.

3.3.2.1 Tokenization

Tokenization splits a text into tokens using nonalphabetic (alphanumeric) characteristics. For example, for the email subject "RE: information requested for PAT test", tokenization gives the tokens "'Re', 'Informations', 'requested, 'For', 'PAT', 'test'". The occurrence of each token in the given subject is one.

3.3.2.2 Stemming

Stemming is the process of reducing words to their root word[1]. One algorithm that can be used to identify the root word of a token is porter stemmer. For the previous example, porter stemmer gives "information," "request," "For," "PAT," "test".

3.3.2.3 Stop word removal

Stop words are the most common words in any language. Those words are widely used words and do not add much information to the content of the text words like "the," "a," "an," "in," "is," "me," "me," "myself," "we," "our," "ours,"...To give much emphasis or more focus to the critical information, we removed those stop words in the email's subject.

3.3.3 Feature Extraction

From the collected and preprocessed dataset, we extracted features that would help to detect spam emails. The extracted features are categorized as new and existing. The ***new features*** are features that we added for this research to use for spam detection. The details of these new features are described in Section 3.2. The ***existing features*** are features used in other studies [6] by previous researchers for email spam detection. The summary of the features is shown in Table 3.3. All features are computed (per sender address and per six hours).

Table 3.3 Existing and New features.

Feature	Feature Type	Metrics
New	Number of recipients	Count of recipient per sender per six hours.
	Number of emails sent	Count of email sent per sender per six hours.
	Time interval of email sent	Mean of time interval per sender per six hours.
	Number of characters in the sender address	Count of characters in the sender address.
	Similarity between email subjects within sender.	Similarity score between email subjects per sender per six hours per receiver.
	Entropy value between email subject within sender.	Entropy value between email subjects per sender per six hours per receiver.
Existing	Number of repetitive words	Mean of Count of repetitive words from the subject of the email.
	Number of nouns	Mean of Count of nouns from the subject of the email.
	Number of words	Mean of Count of words from the subject of the email.
	Number of unique words	Mean of Count of unique words from the subject of the email.
	Number of question marks	Mean of Count of question marks from the subject of the email
	Number of capitalized words	Mean of Count of capitalized words from the subject of the email.

3.3.4 Feature Selection

For feature selection, we use the Pearson Correlation Matrix. The reason why we select these techniques is Pearson Correlation Matrix this matrix helps us see the strongly correlated features and choose one from the two features by seeing the correlation coefficient value.

3.3.5 Model Building

Before building a model, we split the dataset for training and testing. For this, we use ***10-fold cross-validation techniques*** commonly used by various ML techniques and other related research works to minimize bias in our results and variances in estimated results. We used four algorithms K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest for implementing our newly computed features. These algorithms are selected because they are used in several previous studies [2][6][10][12][14]. These algorithms usually give good results by reducing the false positives and false negatives.

3.3.6 Evaluation and Feature Ranking

3.3.6.1 Evaluation (Performance metrics)

To assess the impact of the newly introduced features, we used standard evaluation metrics: precision, recall, F1-score, FPR(False Positive Rate), ROC(Receiver Operating Characteristic), and AUC(Area Under the Curve).

3.3.6.2 Feature Ranking

To assess the importance of each feature, we ranked the features using the Information Gain and Gain Ratio algorithm. The reason for selecting those algorithms are explained below.

1. ***Information Gain*** :- **IG** is used to measuring relevant features from the comprehensive lists. By ranking the relevance of the feature, helps to see how our newly added features are more useful for the performance improvement of this email spam detection technique.

2. ***Gain Ratio***:- **GR** is the modification of IG and is used to reduce the biases of IG [6]. We have ranked the features using this algorithm to see how our newly added features are more helpful in improving this email spam detection technique.

4

EXPERIMENT

This section explains the dataset's characteristics, experimental setup, and the performance metrics used for the empirical experiment.

4.1 Dataset

For our experiment, we collected two months (February and March 2021) of spam email and 21 days of non-spam email data from ethiotelecom. The total number of data collected is shown in Table 4.1. Each instance of the dataset contains aggregated records per sender per six hours.

From the dataset, we extracted twelve features (six new and six existing). The extracted features and their data types are shown in Table 4.2.

Table 4.1 Characteristic of the dataset.

Type	Spam	Non-Spam	Total
Total # of emails	**4397**	**3701**	**8098**
Total # of users	**734**	**998**	**1732**

Table 4.2 Features and their datatypes.

Feature	Dataset
Number of recipients	Numeric
Number of emails sent	Numeric
Duration	Numeric
Character length	Numeric
Similarity	Numeric
Entropy	Numeric
Number of repetitive words	Numeric
Number of nouns	Numeric
Number of words	Numeric
Number of unique words	Numeric
Number of question marks	Numeric
Number of capitalized words	Numeric

4.2 Experimental setup

We conducted two sets of experiments to evaluate the impact of the newly added features in email spam detection: baseline and proposed. The baseline experiment uses the dataset with the existing features to build a model and detect email spam. The existing features are content-based features prepared by Ammara Zamir et al.[6].

The proposed approach uses the dataset containing both existing content-based features and the new features proposed in this study to build a model and detect spam emails. The overall experiment setup is shown in Figure 4.1. In the first set of experiments, we used the content-based features prepared by Ammara Zamir et al.[6] to build and evaluate email spam detection models. We used the new dataset to build and evaluate the email spam detection models in the second set of experiments. For both types of features, we use the same dataset in Table 4.1.

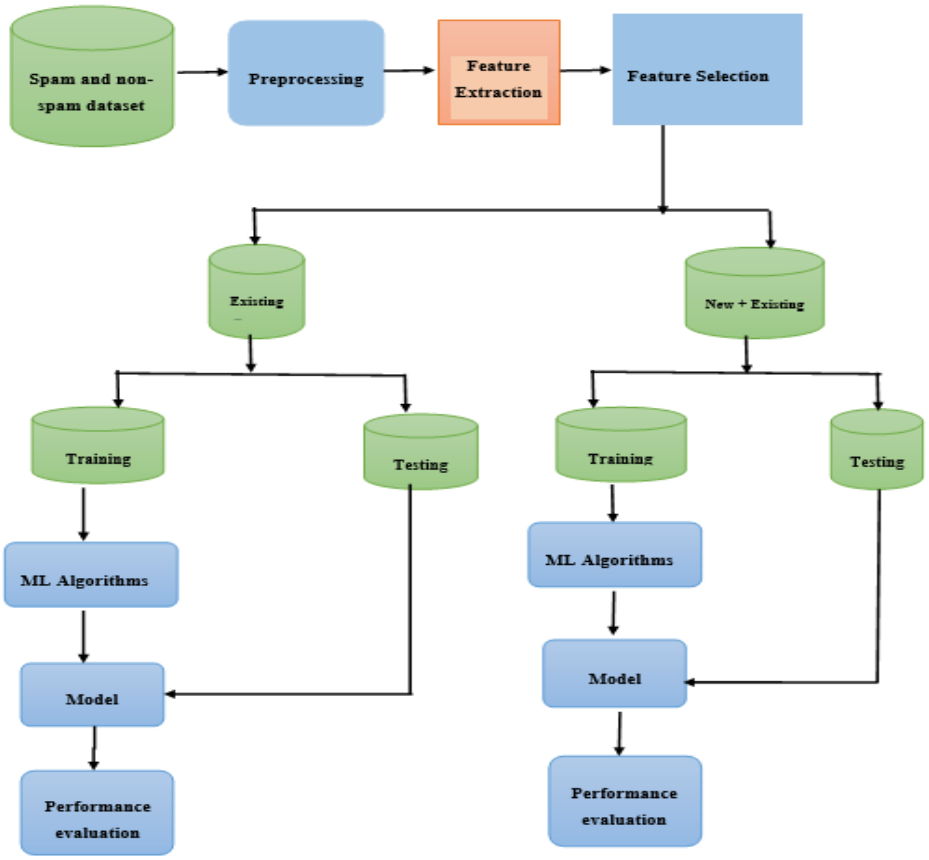


Figure 4.1 Experimental Design

4.3 Feature Selection

Using different feature selection techniques, we select the most valuable features for our classification. Sometimes, all parts are not necessary for prediction, so knowing the helpful features is essential.

For this research, we used the Pearson Correlation Matrix for feature selection purposes.

4.3.1 Pearson Correlation Matrix

We used a correlation matrix to identify features that strongly correlate and select a representative feature. This study considered features with a correlation value equal to or greater than 0.8 as strongly correlated.

Table 4.3 Correlation Matrix

	Sent email	Number of receivers	Length of address	Duration	Similarity	Entropy	Number of words	Number of repititive words	Number of nouns	Number of unique words	Number of capitalized words	Number of question marks
Sent email	1	0.12	0.09	-0.05	0.01	0.2	-0	0.17	0.11	-0.1	0.02	0.08
Number of receivers	0.12	1	0.06	0.01	-0.01	0	0.02	-0.2	-0.1	0.07	0	-0.02
Length of address	0.09	0.06	1	-0.02	0.04	0.1	0.13	-0	0.09	0.1	0.06	-0.02
Duration	-0.05	0.01	-0.02	1	-0.01	0	-0	-0.1	-0	0.02	-0.01	0
Similarity	0.01	-0	0.04	-0.01	1	0	0.01	0.04	0.02	0.01	-0.01	-0.01
Entropy	0.22	0.01	0.06	0.01	0.04	1	0.01	0.14	0.13	-0.1	-0.06	0
Number of words	-0.03	0.02	0.13	-0.01	0.01	0	1	0.02	0.29	0.9	0.24	0.03
Number of repetitive words	0.17	-0.2	-0.02	-0.07	0.04	0.1	0.02	1	0.18	-0.3	0.01	0.03
Number of nouns	0.11	-0.1	0.09	-0.02	0.02	0.1	0.29	0.18	1	0.2	0.09	0.02
Number of unique words	-0.09	0.07	0.1	0.02	0.01	-0.1	0.9	-0.3	0.2	1	0.22	0.02
Number of capitalized words	0.02	0	0.06	-0.01	-0.01	-0.1	0.24	0.01	0.09	0.22	1	0.08
Number of question marks	0.08	-0	-0.02	0	-0.01	0	0.03	0.03	0.02	0.02	0.08	1

Below Table 4.4 shows us uncorrelated features after the correlation matrix that we used for model building. Previously before this correlation matrix, there were twelve features. After this correlation matrix, eleven features are uncorrelated.

Table 4.4 Uncorrelated features

NO	List of uncorrelated features
1	Length of address
2	Number of repetitive words
3	Number of receivers
4	Duration
5	Sent emails
6	Entropy value
7	Number of nouns
8	Number of words
9	Similarity score
10	Number of question marks
11	Number of capitalized words

4.4 Model Building

Below are the details of the machine learning techniques applied to the selected data set. To get a better predictive model, we implemented the below four machine learning algorithms.

Support Vector Machine

Support Vector Machine (SVM) algorithm. It is a supervised classification algorithm. Support Vector Machine uses labeled data for classification. It is a more accurate method [4]. Support Vector Machine is more suitable for classification purposes. Support Vector Machine works well with high-dimensional datasets. Support Vector Machine had been used as a spam detection algorithm in many previous works, such as in [4][6][14]. This algorithm is grounded on the notion of structure minimization of risk, which intends at identifying the hyper-plane which divides the spam and non-spam perfectly. Points lying on the hyper-plane are known as support vectors that are utilized in the decision-making function. Apart from the hyperplane, there are two marginal planes (support vectors). Also, the marginal lines pass one of the nearest spam and non-spam data. This algorithm is used for solving both classification and regression. The classes are linearly separable. It is suitable for applications that involve separable datasets and datasets that are not separable [2].

The Support Vector Machine aims to find a maximum marginal hyperplane(MMH) to divide the datasets into classes. It can be done in the below two steps [29].

- Support Vector Machine will generate hyperplanes iteratively that segregate the classes in the best way.
- it will select the hyperplane that separates the classes correctly.

K Nearest Neighbor Algorithms

K Nearest Neighbor (**KNN)** is a supervised classification algorithm. It searches similar data from the with K nearest instances. Where K is the number of Nearest Neighbors that we want to select. K is the most critical parameter in a text categorization system. Using this classification process, the first k nearest documents to the test one in the training set are determined. Accordingly, the prediction is performed. Some classes may have excessive samples than other classes. However, the system performance is susceptible to the choice of the parameter k [15]. it classifies the new data based on the similarity measures. K Nearest Neighbor is a lazy learning algorithm because it does not have a specialized training phase and during classification uses all the data for training [29]. K Nearest Neighbor algorithm calculates Euclidian distance

and ranks the samples according to the distance [2]. Helps to predict based on the distance between the new email and the email in the training set.

Using the two datasets, which is existing and new plus existing K Nearest Neighbor algorithm follows the below steps for during model building [29].

1. First, Load the training and testing dataset.
2. Then, choose the value of K, which is the nearest data point. K can be any integer.
3. Calculate the distance between test and training data with the help of Euclidean distance.
4. Based on the distance value, sort them in ascending order.
5. Choose the top K rows from the sorted array.
6. Finally, assign the class to this test point from the most frequent class of those rows.

Logistic Regression

Logistic Regression is a supervised machine learning algorithm which is based on the probability concept and its cost function lies between 0 and 1 [2] which means used to predict from a large volume of features. It's used to predict categorical variable with the help of dependent variable. The logistic function that is a sigmoid function is an 'S' shaped curve that takes any real values and converts them between 0 to 1. If the output given by a sigmoid function is more than 0.5, the output is classified as spam & if is less than 0.5, the output is classified as non-spam in our research. If the graph goes to a negative end, then y predicted will be 0 and vice versa. In this method sigmoid function is used to model the data as shown below [2].

$$g(z) = \frac{1}{1+e^{-z}} \quad (4.1)$$ (Adopted from [2])

$$y = \frac{e^{(b0+b1*x)}}{(1 + e^{(b0+b1*x)})}$$

(4.2) (Adopted from [2])

Where :

y shows the predicted value

x shows the input feature

bo shows the bias

b1 shows the coefficient for the feature

Random Forest

Random Forest is mainly used for classification problems. A random Forest algorithm creates decision trees, gets each prediction, and finally selects the best solution using voting. It avoids overfitting to produce better results. Random Forest is used to model the non-linear class boundaries [6]. The designed strategy used in Random Forest is divide and conquer. It forms several decision trees, and each decision tree is trained by selecting any random subset of attributes from the whole predictor attribute set.

Random Forest runs efficiently in an extensive range of data than a single decision tree. It also handles the missing values inside the data set used for training. Handle the unbalanced data set by using Random Forest is difficult. The Random Forest algorithm is very flexible and possesses very high accuracy. It maintains good accuracy even a significant proportion.

Using the two datasets, which is existing and new plus existing Random Forest algorithm follows the below steps for during model building:-

1. From a given dataset, select random samples for decision tree (DT) construction purposes.

2. A decision tree is constructed for every sample. After getting the prediction result from every decision tree, voting will be performed for every predicted result.

3. Using the voting system, the most voted prediction result will be selected as the final prediction result.

4.5 Evaluation Metrics

To evaluate the performance of the proposed approach, we used metrics derived from a confusion matrix.: Accuracy, Precision, Recall, F1-score, false-positive rate (FPR), and ROC. The confusion matrix contains four classes: True Negative (TN), True Positive(TP), False Negative(FN), False Positive(FP) (see Table 4.3).

Table 4.3 Description of a confusion matrix

		Actual value	
		Spam	**Non-Spam**
Predictive value	**Spam**	TP (The actual value was spam and correctly predicted as spam.)	FP (The actual value was non-spam and incorrectly predicted as spam.)
	Non-Spam	FN (The actual value was spam and incorrectly predicted as non-spam.)	TN (The actual value was non-spam and correctly predicted as non-spam.

Accuracy

Accuracy is used as a performance measure in the domains of information retrieval and data mining. Accuracy helps us see the correctness of the prediction. It's mainly focuses on the TP and TN values. It helps to know the positive predicted values from the overall dataset. Accuracy is computed using equation 4.4.

$$\textbf{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{4.4}$$

False-positive rate

It is one of the performance evaluation metrics that help to evaluate the positives value from the overall negative values. *We* aim to increase the TN and TP values for email spam detection and reduce FP and FN values for spam detection, so these performance metrics help evaluate our aim. FPR is computed using equation 4.5.

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \tag{4.5}$$

Precision

Precision mainly focuses on the FP value. Since spam detection aims to decrease FP values, these metrics help to see these values. Precision is calculated using equation 4.6.

$$Precision = \frac{TP}{TP+FP} \tag{4.6}$$

Recall

Recall helps to see from the actual positive values how many of them are predicted as positive. It is mainly focused on FN values. The recall value of each predicted model is calculated using equation 4.7.

$$Recall = \frac{TP}{TP+FN} \tag{4.7}$$

F-measure

F1-Score is the harmonic mean value of precision and recall. The F1-Score value of each predicted model is calculated using equation 4.8.

$$\boldsymbol{F1 - Score = \frac{2*(precision*recall)}{(precision+recall)}} \tag{4.8}$$

ROC

The receiver operating characteristic(ROC) is used how well a classifier works. The ROC curve is plotted using false positive rate and true positive rate as x and y-axis with different threshold values. The area below this curve is called the area under the curve(AUC).

4.6 Feature Ranking

For this research, we used Information Gain and Gain Ratio algorithms for feature ranking purposes, which helps to see the feature importance by ranking the top important features.

4.6.1 Information Gain

It is a feature ranking algorithm used to extract useful features from the selected one. It calculates the information of an attribute given about a class. It is used to measure the reduction in entropy. The Information Gain is calculated for each feature in the dataset. The features that has the largest Information Gain is selected to split the dataset. Generally, a larger gain indicates a smaller entropy. We choose this algorithm because other than the functions stated above, and it helps to rank the most valuable features from the entire feature lists using the dataset. It helps to see which parts are at the top for comparing the new and existing features.

4.6.2 Gain Ratio

Gain Ratio(GR) is the modification of Information Gain and is used to reduce the biases of Information Gain. Gain Ratio is used to normalize the value of Information Gain by using intrinsic information. The reason we choose this algorithm is that other than the functions stated above, it helps to rank the most valuable features from the total feature lists using the dataset, and it helps to see which parts are at the top for comparing the new and existing features.

5

RESULT AND DISCUSSION

This chapter presents and discusses the results of the experiment described in Chapter 4. A predictive model built using existing and new plus existing feature datasets were used to assess the contribution of the new features identifying spam email from non-spam. The results, along with the discussion, are presented below.

5.1 Result

5.1.1 Performance of augmented features

The first research question, RQ1: *To what extent do the newly added features improve the performance of email spam detection*? aims to assess the impact of the newly added features in improving email spam detection. The results of the experiment conducted to answer this question show that the augmented features, i.e., new plus existing, improved email spam detection by an overall average F1 score of 12.1%. The results of the baseline and augmented features for the four machine learning algorithms are shown in Table 5.1 and Figure 5.1.

The F1-score of the email spam detection is improved by 6.6%, 9.9%, 20.7%, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively. The most significant improvement in F1-score, 20.7%, is observed using the logistic regression algorithm. The F1-score of the existing feature dataset is lower than 90% in all algorithms. Compared to the other algorithm, F1-score improvement is relatively low for K Nearest Neighbor while using the existing feature. False-positive rate lets us see from actual non-spam values how many of them are predicted as spam. The FPR has an improvement while using the augmented dataset with Random Forest.

The augmented feature dataset also shows improvement in accuracy for all algorithms. The improvements in accuracy are 5.49%,8.14%,14.2%, and 9.52% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively. This shows that the newly added features help to improve email spam detection. The highest accuracy and F1-score are observed while using Random Fores followed by Logistic Regression. The lowest accuracy and F1-score are observed for K Nearest Neighbor. This implies that machine learning algorithms in email spam detection also have a significant role in performance improvement.

Table 5.1 Overall performance evaluation result.

Algorithm	Dataset	Accuracy(%)	Δ Accuracy	FPR(%)	AUC(%)	Precision(%)	Recall(%)	F1(%)	Δ F1
K Nearest Neighbor	Augmented	**93.19**	**5.49**	**6.5**	**93.9**	**91.3**	**92.8**	**92**	**6.6**
	Baseline	**87.7**		**10.2**	**93.3**	**86**	**84.9**	**85.4**	
Support Vector Machine	Augmented	**95.09**	**8.14**	**2.9**	**94.7**	**95.9**	**92.4**	**94.1**	**9.9**
	Baseline	**86.95**		**9.4**	**86.3**	**86.5**	**82**	**84.2**	
Logistic Regression	Augmented	**93.65**	**14.2**	**3.9**	**98.1**	**94.5**	**90.3**	**92.4**	**20.7**
	Baseline	**79.45**		**7.3**	**88.6**	**86.1**	**61.5**	**71.7**	
Random Forest	Augmented	**97.63**	**9.52**	**2**	**99.7**	**97.3**	**97.1**	**97.1**	**11.3**
	Baseline	**88.11**		**9.2**	**94**	**87.1**	**84.5**	**85.8**	

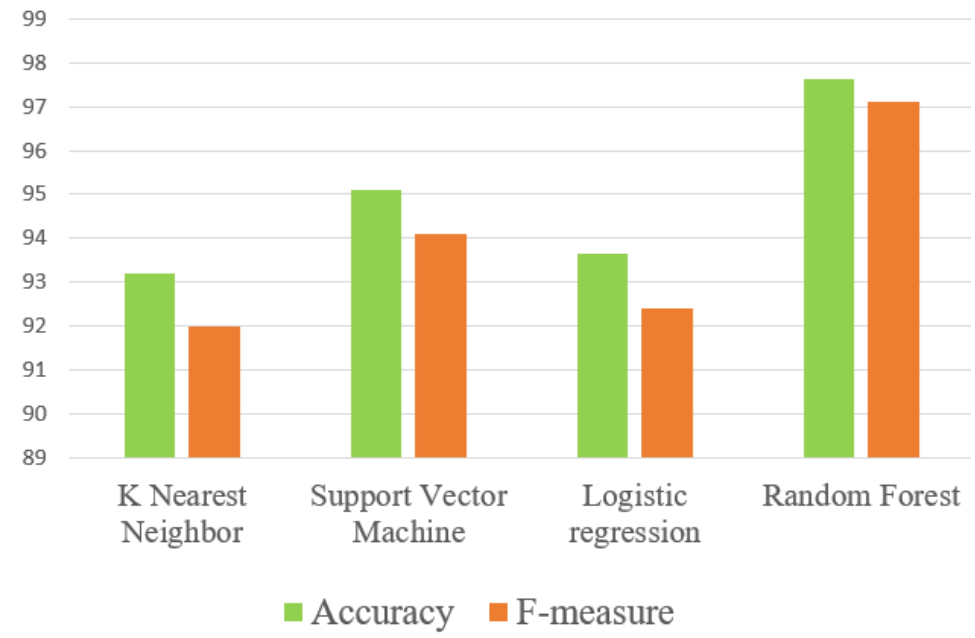


Figure 5.1 The Augmented feature dataset F1-score and Accuracy.

Looking at the AUC curve, a better AUC value is observed in the augmented feature dataset. AUC values of 93.3%, 86.3%, 88.6%, 94% are achieved while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively (see Figure 5.2). A higher AUC value is observed while using Random Forest.

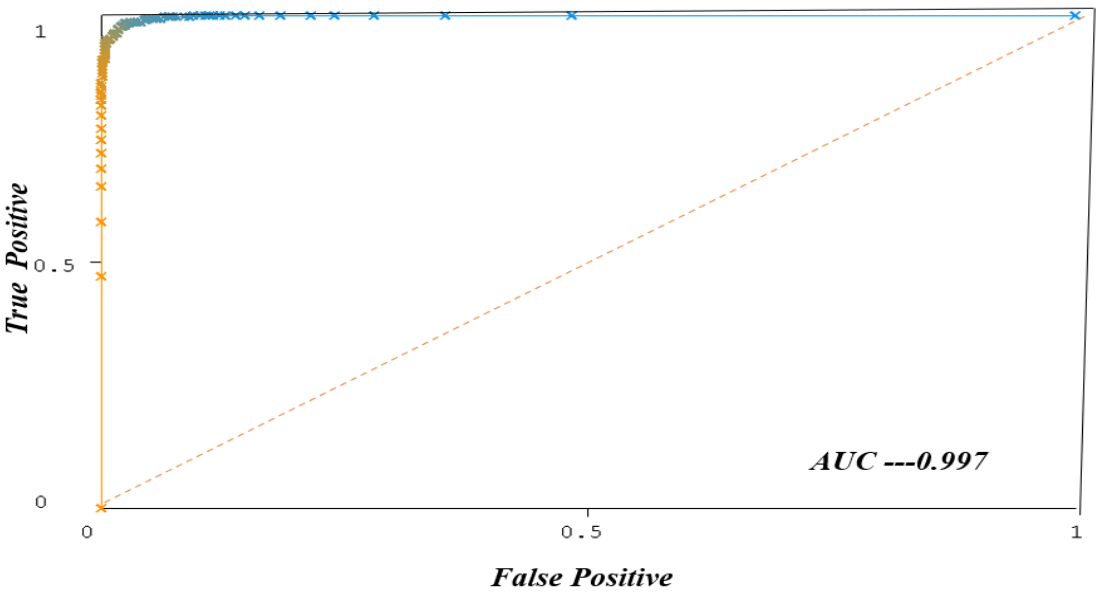


Figure 5.2 AUC for the augmented feature using RF.

5.1.2 Contribution of newly added features

To answer the second research question, RQ2: *Which of the newly added features contribute more to improve email spam detection?* we ranked the augmented feature sets using feature importance ranking algorithms: Information Gain and Gain Ratio. The result shows that the newly added features are ranked in the top six features (Table 5.2). This shows that the newly added features contribute more to the improvement of email spam detection.

While using the Information Gain algorithm, *length of address, the number of receivers, duration, sent emails, and entropy* are ranked in the top six based on their importance to email spam detection. The *number of question marks* feature is the least important from the total of 11 features used in the detection of email spam.

Both Gain Ratio and Information Gain algorithms give similar results (see Table 5.2). Both algorithms rank *length of address, the number of receivers, duration, sent emails, and entropy* in the top six with minor variations in order. Most of the existing features are found to be less important to the detection of spam emails. From the existing features, the highest rank is observed by *number of capitalized words* feature while using Gain Ratio and *number of Nouns* feature while using Information Gain algorithms. *number of Capitalized words* is ranked fifth while n*umber of nouns* is ranked seventh.

Table 5.2 IG and GR Ranking

Information Gain Ratio(IG)		**Gain Ratio(GR)**	
Feature	**Rank**	**Feature**	**Rank**
Length of address	0.46	Number of Repetitive words	0.45
Number of Repetitive words	0.35	Number of recipients	0.18
Number of recipients	0.33	Length of address	0.17
Duration	0.28	Duration	0.16
Sent email	0.17	Number of capitalized words	0.13
Entropy	0.13	Entropy	0.09
Number of nouns	0.11	Sent email	0.09
Number of words	0.10	Number of nouns	0.06
Similarity	0.06	Similarity	0.05
Number of question marks	0.01	Number of words	0.04
Number of capitalized words	0.00	Number of question marks	0.01

5.2 Discussion

The overall aim of this research is to improve the performance of email spam detection. In this regard, we proposed adding six additional features based on content, semantics, and entropy that take into consideration the sender. The results show that the proposed features improved the detection of spam emails.

Improved email spam detection implies increased true positive (TP) and true negative (TN) values; in contrast, the false positive (FP) and false-negative (FN) values are decreased for the respective machine learning algorithms.

In all experiments F1-score is improved while using the augmented (new+existing) dataset. This shows that the newly added features impact the F1-score. The augmented feature dataset improves the false-negative rate by 2%. When we see the delta value of the F1-score, the new + existing feature dataset improves email spam detection in the four algorithms by an overall average of 12.1%.

To get insight for further implementation suggestions, we compare the output of four algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. We use six performance metrics: accuracy, F1-score, FPR, precision, recall, and AUC. The result shows that Random Forest is better than the other three machine learning algorithms. The lowest performance is observed for K Nearest Neighbor. From this, one can conclude that the performance of email spam detection is also dependent on the selected machine learning algorithm.

The experiment also shows that the feature contributing more to this email spam detection improvement is the number of emails sent, number of receivers, duration, and entropy. The least important feature is the *number of capitalized words* and the *number of question marks* while using Information Gain and Gain Ratio algorithms, respectively.

6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Many organizations use email for various activities. However, the valuable time of these organizations is wasted by unwanted (spam) emails sent from different sources. To solve this problem, many have proposed to use different email spam detection approaches. A spam detection approaches help the business prevent unnecessary emails from reaching their inboxes and preventing any consequential harm to the organization. Spam detection mechanism helps save the business resource, prevent the industry from different attacks, improve employee productivity, and protect company revenue and data. In the state of art email spam detection, various researchers used machine learning-based approaches. However, the features used in the machine learning-based strategies focus only on content-based features and have relatively low performance. To improve the performance, in this study, we propose to augment the existing features with sender centric features: number of emails, duration, character length of the sender address, number of recipients, the similarity between emails, and entropy value within the sender subject.

To assess the impact of the newly added features on improving email spam detection, we conducted two experiments, one with existing features only, our baseline features, and another with New + Existing features. We use four predictive models built using four commonly used machine learning algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. After the experiment, we compare the performance of the augmented feature dataset with the baseline features dataset using six performance metrics.

The result of the experiments show that the new + existing dataset improves the performance of email spam detection in all experiments. The email spam detection accuracy is improved by 5.4%, 8.14%, 14.2%, and 9.52% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest while using augmented feature dataset. The F1-score of the email spam detection is also improved by 6.6, 9.9, 20.7, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest, respectively. Among the machine learning algorithms, Random Forest performs better in all the experiments.

To see which of the features contributed more to the improvement of email spam detection, we ranked the features using feature ranking algorithms GR and IG. The result shows that the number of emails sent, duration, number of receivers, number of characters in sender address, and similarity are in the top six important features. The *number of capitalized words* and *number of question marks*, from the existing features, are the least essential features. Hence, we can conclude that the newly added features play a crucial role in email spam detection.

6.2 Future Work

In this thesis work, email subjects are used to extract different features. Email subjects provide limited information about the email when compared to the body of the email. Hence, we believe that if the email body is used more improvement could be achieved. In the future, we plan to investigate the impact of using email body along with the email subject in improving detection of email spam.

Reference

[1] T. Sultana, "Email-based Spam Detection," *Int. J. Eng. Res.*, vol. V9, no. 06, pp. 135–139, 2020, doi: 10.17577/ijertv9is060087.

[2] S. Nandhini and D. J. Marseline, "Performance Evaluation of Machine Learning Algorithms for Email Spam Detection," *Int. Conf. Emerg. Trends Inf. Technol. Eng. ic-ETITE 2020*, pp. 1–4, 2020, doi: 10.1109/ic-ETITE47903.2020.312.

[3] H. Bhuiyan, A. Ashiquzzaman, T. I. Juthi, S. Biswas, and J. Ara, "A Survey of Existing E-Mail Spam Filtering Methods Considering Machine Learning Techniques," *Glob. J. Comput. Sci. Technol. Softw. Data Eng.*, vol. 1, no. 2, 2018.

[4] S. Suryawanshi, A. Goswami, and P. Patil, "Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers," *Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019*, pp. 69–74, 2019, doi: 10.1109/IACC48062.2019.8971582.

[5] M. Zhiwei, M. M. Singh, and Z. F. Zaaba, "Email Spam Detection : a Method of Metaclassifiers Stacking," *Int. Conf. Comput. Informatics*, no. 200, pp. 750–757, 2017.

[6] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal, and A. U. Akram, "A feature-centric spam email detection model using diverse supervised machine learning algorithms," *Electron. Libr.*, vol. 38, no. 3, pp. 633–657, 2020, doi: 10.1108/EL-07-2019-0181.

[7] M. V. Madhavan, S. Pande, P. Umekar, T. Mahore, and D. Kalyankar, "Comparative analysis of detection of email spam with the aid of machine learning approaches," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, doi: 10.1088/1757-899X/1022/1/012113.

[8] N. Parmar, A. Sharma, and H. Jain, "Email Spam Detection using Naïve Bayes and Particle Swarm Optimization," vol. 6, no. 10, pp. 367–373, 2020.

[9] D. M. Ablel-Rheem, A. O. Ibrahim, S. Kasim, A. A. Almazroi, and M. A. Ismail, "Hybrid feature selection and ensemble learning method for spam email classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1.4 Special Issue, pp. 217–223, 2020,

doi: 10.30534/ijatcse/2020/3291.42020.

[10] P. Sharma and U. Bhardwaj, "Machine learning based spam Email detection," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 3, pp. 1–10, 2018, doi: 10.22266/IJIES2018.0630.01.

[11] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[12] M. Vinodhini, D. Prithvi, and S. Balaji, "Spam Detection Framework using ML Algorithm," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 5326–5329, 2020, doi: 10.35940/ijrte.f1120.038620.

[13] R. Talaei Pashiri, Y. Rostami, and M. Mahrami, "Spam detection through feature selection using artificial neural network and sine–cosine algorithm," *Math. Sci.*, vol. 14, no. 3, pp. 193–199, 2020, doi: 10.1007/s40096-020-00327-8.

[14] M. Zhou, S. Zhang, Y. Qiu, H. Luo, and Z. Wu, "Entropy-based spammer detection," *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3240876.3240901.

[15] D. Mallampati and N. P. Hegde, "A Machine Learning Based Email Spam Classification Framework Model: Related Challenges and Issues," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 4, pp. 3137–3144, 2020, doi: 10.35940/ijitee.d1561.029420.

[16] L. Baoli, Y. Shiwen, and L. Qin, "An Improved k -Nearest Neighbor Algorithm," *Proc. 20th Int. Conf. Comput. Process. Orient. Lang.*, no. July, 2003.

[17] N. Hussain, H. T. Mirza, and I. Hussain, "Detecting Spam Review through Spammer's Behavior Analysis," *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.*, vol. 8, no. 2, pp. 61–71, 2019, doi: 10.14201/adcaij2019826171.

[18] M. Lakshmi, "Email Security Using Spam Mail Detection and Filtering Network System," *Int. J. Eng. Tech.*, vol. 4, no. 1, pp. 551–554, 2018, [Online]. Available: http://www.ijetjournal.org.

[19] V. Christina, S. Karpagavalli, and G. Suganya, "Email Spam Filtering using Supervised Machine Learning Techniques," vol. 02, no. 09, pp. 3126–3129, 2010.

[20] W. S. Awad and W. M. Rafiq, "Improving Spam Email Filtering Systems Using Data Mining Techniques," pp. 43–72, 2020, doi: 10.4018/978-1-7998-2418-3.ch003.

[21] W. Awad, "Machine Learning Methods for Spam E-Mail Classification," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 173–184, 2011, doi: 10.5121/ijcsit.2011.3112.

[22] T. Zar Phyu and N. N. Oo, "Performance comparison of feature selection methods," *MATEC Web Conf.*, vol. 42, pp. 2–5, 2016, doi: 10.1051/matecconf20164206002.

[23] K. Pawar and M. Patil, "Spam Filtering Security Evaluation Framework Using SVM, LR and MILR," *Int. J. Comput. Technol.*, vol. 3, no. 2/3, pp. 19–27, 2016, doi: 10.5121/ijcax.2016.3302.

[24] M. Rubin Julis and S. Alagesan, "Spam detection in sms using machine learning through text mining," *Int. J. Sci. Technol. Res.*, vol. 9, no. 2, pp. 498–503, 2020.

[25] P. Hoonakker, P. Carayon, and N. Bornø, "Spamming , spoofing and phishing Email security : A survey among end-users," *Int. Ergon. Assoc. 2009 17th World Congr. Ergon.*, no. August, pp. 1–7, 2009.

[26] M. Iqbal, M. M. Abid, M. Ahmad, and F. Khurshid, "Study on the Effectiveness of Spam Detection Technologies," *Int. J. Inf. Technol. Comput. Sci.*, vol. 8, no. 1, pp. 11–21, 2016, doi: 10.5815/ijitcs.2016.01.02.

[27] D. Scuse and P. Reutemann, "WEKA Experimenter Tutorial for Version 3-5-4, *Test*,2006"[Online].Available: http://qa.debian.org/watch/sf.php/weka/ExperimenterTutorial-3.5.4.pdf. [Accessed: 09-Jun-2021].

[28] " How scammers use phishing attacks to steal, exploit company data" [Online]. Available: https://www.sbnonline.com/article/how-scammers-use-phishing-attacks-to-steal-exploit-company-data/. [Accessed: 02-Aug-2021]

[29] “Classification Algorithms Random Forest.” [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_wit

h_python_classification_algorithms_random_forest.htm. [Accessed: 09-Jun-2021].

[30] "Classification Algorithms Logistic Regression." [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm. [Accessed: 09-Jun-2021].

[31] N. Govil, K. Agarwal, A. Bansal, and A. Varshney, "A Machine Learning based Spam Detection Mechanism," no. Iccmc, pp. 954–957, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-000177.

[32] A. G. Asuero, A. Sayago, and A. G. González, "The correlation coefficient: An overview," *Crit. Rev. Anal. Chem.*, vol. 36, no. 1, pp. 41–59, 2006, doi: 10.1080/10408340500526766.

[33] N. Mirza, B. Patil, T. Mirza, and R. Auti, "Evaluating efficiency of classifier for email spam detector using hybrid feature selection approaches," *Proc. 2017 Int. Conf. Intell. Comput. Control Syst. ICICCS 2017*, vol. 2018-January, pp. 735–740, 2017, doi: 10.1109/ICCONS.2017.8250561.

[34] S. Negi and Rekha , "A Review on Different Glaucoma Detection," *Int. J. Eng. Trends Technol.*, vol. 11, no. 6, pp. 2–7, 2015.

[35] B. A. Kamoru, A. Bin Jaafar, M. A. A. Murad, E. O. Ernest, and M. B. A. Jabar, "Spam Detection Approaches and Strategies: A Phenomenons," *Int. J. Appl. Inf. Syst.*, vol. 12, no. 9, pp. 13–18, 2017, doi: 10.5120/ijais2017451728.

[36] "Mail flow and the transport pipeline" [Online]. Available: https://docs.microsoft.com/en-us/exchange/mail-flow/mail-flow?view=exchserver-2019. [Accessed: 27-Jun-2021].

[37] " What's new in Exchange Server" [Online]. Available: https://docs.microsoft.com/en-us/exchange/new-features/new-features?view=exchserver-2019. [Accessed: 28-Jun-2021].

[38] J. Thomas, P. Vinod, and N. S. Raj, "Towards spam mail detection using robust feature evaluated with feature selection techniques," *Int. J. Eng. Technol.*, vol. 6, no. 5, pp. 2144–2158, 2014.

[39] "Weka Tutorial" [Online]. Available: What's new in Exchange Server" [Online]. Available: https://www.tutorialspoint.com/weka. [Accessed: 24-April -2021].

[40] D. Scuse and P. Reutemann, “WEKA Experimenter Tutorial for Version 3-5-4,” Test, 2006,[Online]. Available:https://weka/ExperimenterTutorial-3.5.4.pdf.

[41] "Cross-validation in machine learning "[Online]. Available:https://tutorialspoint.dev/computer-science/machine-learning/cross-validation-machine-learning. [Accessed: 14-May -2021].

APPENDIX

Email Spam Detection Using Content, Semantic, and Entropy-based features

Tigist Beyene Mamo
Addis Ababa Institute of Technology
Addis Ababa University, Addis Ababa, Ethiopia
tigistbeyene1719@gmail.com

Surafel Lemma Abebe
Addis Ababa Institute of Technology
Addis Ababa University, Addis Ababa, Ethiopia
surafel.lemma@aait.edu.et

Abstract— The spam detection technique helps the business prevent unnecessary emails from reaching inboxes and preventing any consequential harm to an organization or user. To train the machine learning algorithms, several researchers use features extracted from the email. These features, however, capture only the content and are computed per email message. However, information aggregated per sender, i.e., content, entropy, and spam email similarity per sender, are not studied. In this study, we propose to use additional features that capture the content, similarity, and entropy of emails sent by the same sender. In this regard, we extracted six new features that could help to improve spam email detection. The six features are number of emails, duration, character length of the sender address, number of recipients, the similarity between emails, and entropy value within the sender subject. To build the prediction model, we used four machine learning algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. The proposed approach is evaluated using a dataset collected from ethiotelecom. The results show that the dataset augmented with the new features improves email spam detection performance. The F1-score of email spam detection is improved by 6.6%, 9.9%, 20.7%, and 11.3% using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. The overall improvement is 12.1%. Among the Four algorithms used to build the predictive models, Random Forest performs better in detecting spam emails. We computed feature importance using the Information Gain and Gain Ratio algorithms to see which features helped to improve email spam detection. The result shows that the new features, length of address, the number of receivers, duration, sent emails, and entropy, are in the top six ranks. This indicates that the newly introduced features contributed to the improvement seen in email spam detection.

Index-terms— Email Spam detection, Classification, Supervised Machine Learning, proposed new features, Feature importance.

.

I. INTRODUCTION

Telecom service providers use email services for their day-to-day operational activities. By using this application, they send and receive sensitive data. Hence, email security becomes more critical in any operator/organization. Email security refers to protecting email users from various attacks [9]-[11]. There are many types of email security threats such as Viruses,Phishing,Man-In-The-Middle, Eavesdropping, Dictionary attacks, Spam, Denial of service attacks [9]-[11]. One of the mechanisms to secure email users is the use of proper email spam filtering techniques.

Spam emails are annoying to most users, and users receive emails without the user s knowledge. The growing problem of email spam motivates the emergence of email spam detection and filtering technique. Email spam detection techniques can broadly be classified as rule-based and machine learning-based techniques. The rule-based techniques work by setting a set of rules for classification. Rules help to detect and filter incoming emails on the email server. However, the spammer's behavior is not statics, and their character change frequently. Hence, these techniques are not effective in filtering spams with new behavior. Rule-based techniques have more chances to increases the false positive and false negative values. The impact of false positives is that the technique considers essential emails as spam and filtered them out. On the other hand, False negatives see spam emails as normal emails that will affect the user's activity. The impact of false-positive value to the user is that a particular email may be essential for the user because of the predicted result user may lose that email.

On the other hand, the user may be attacked by that spam email for the false-negative result since that email is spam. Many have suggested using machine learning approaches to detect and filter emails[11]. Machine learning techniques are the most widely used techniques for spam detection/filtering [11]. These techniques use data rather than predefined rules. The machine learning algorithms used for spam detection are Naïve Bayes, support vector machines, neural networks, K-nearest neighbor, rough sets, logistic

regression, random forest [11]. These techniques use datasets with independent variables as input and predict the dependent variable as the output. In classification, the input dataset is split into two as training and testing. The training dataset is used for model building, while the testing dataset is used to evaluate the model's performance. Machine learning techniques help improve email spam detection using different email features since the features are extracted from the behavior of the spam and non-spam data of that specific operator/organization.

Different researchers use different email features to improve email spam detection using machine learning algorithms. Depending on the selected features, the performance of the machine learning approaches also varies. For this research, we used twelve features; the following features are used in the state-of-the-art number of repetitive words, number of nouns, number of words, number of unique words, number of capitalized words, number of question marks. The remaining features like number of emails sent, duration, number of sender address characters, number of recipients, the similarity between emails, and entropy value (diversification of information) between emails are the newly added features.

This research will build four predictive models using four machine learning algorithms (K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest) for having a better email spam detection model. To assess the impact of the newly added features, we conducted two experiments for the stated datasets and finally compared the performance. The two different datasets are the existing features (features in state of the art) and another with new + existing features (existing and the newly added). By comparing the performance evaluation of the two datasets, we can see how valuable the newly added features are. Using the dataset gathered from ethiotelecom. The results show that the augmented dataset improves the performance of email spam detection in all the scenarios. The F1-score of the email spam detection is improved by 6.6%, 9.9%, 20.7%, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest, respectively.

The rest of the paper is organized as follows. Section II discusses proposed features. Sections III and IV present the approach that we follow for implementing the newly computed features and the experimental setup used to assess the identified metrics. The results of the experiment are presented in Section V. Finally, Section VI concludes the paper and provides future research directions.

II PROPOSED FEATURES

To enhance the detection of email spam, we propose to augment the existing features with content-based, semantic, and entropy-based features. Content-based features aim to capture different aspects of spam email content. On the other hand, the semantics-based feature aims to capture the semantics using similarity measures between subjects of different emails. In contrast, the entropy-based feature captures the diversity of the subject of the email.

Below is a short description of each feature as per the category content-based, similarity-based, entropy-based. And the reason for selecting this feature is in parallel with the characteristic of the spam email.

A. Content-Based

From any organization, email data perspective, content-based features improve email spam detection [6]. It means that using the content of the email its possible to compute relevant features. For this research, using the subject content of the user email, the listed features are computed: the number of recipients of the email, the number of emails sent, an interval of email sent, character length of the sender address, all with a specific period. Below is a short description of these features.

- ***Number of recipients***

We added this feature taking into consideration as an expert by considering the following characteristics of spam and non-spam emails. That is, one spammer may send extensive emails to one user. However, when we see the non-spam users, extensive emails are not sent to one user but rather to different recipients. So, knowing this feature helps to improve email spam detection performance by counting the recipient per sender-specific period.

- ***Number of emails sent***

Counting the number of emails sent per sender per recipient will also help improve the performance of email spam detection. We added this feature as an expert by considering the following characteristics of spam and non-spam emails. One recipient may receive too many emails from the same sender. However, in non-spam emails, one recipient may receive few emails from the same sender per specific period, so knowing this helps improve email spam detection. This feature is computed by counting the number of emails sent per sender per recipient within a specific period. That is, spammers send many emails per day other than the non-spam users.

- ***Time interval of email sent***

Spammers usually repeatedly send several emails in short time intervals. However, legitimate users take a relatively long time. These characteristics, hence, could help distinguish spam email from a non-spam email. This feature is computed by taking the interval between emails for a specific recipient per sender within a specific period. Since the sender aggregates data, the final duration is considered by taking the average of the interval within the recipient per sender.

- ***Number of characters in the sender address***

The sender address of spam emails is usually longer than legitimate emails. Hence, it could serve as a good feature that can differentiate spam emails from legitimate emails. This feature is computed by counting the number of characters in the sender's address.

B. Semantic Based

Semantic similarity helps to check the similarity between two documents. For our study, it checks the content similarity between email subjects of the same recipients.

- ***The similarity score between email subjects by the sender***

This feature is computed within the subjects. Its value evaluates the similarity between subject contents per recipient. For example, let X is the sequence of email subjects within the same sender, i.e., X = X1, X2, ..., Xn. The number times the similarity scores are calculated using eq 3.3. the similarity score between two subjects is calculated using eq 3.2 .then we consider the mean similarity score value for the recipients.

$$Cos\ \theta = \frac{\sum_{i=1}^{n} X1i\ X2i}{\sqrt{\sum_{i=1}^{n} X1i^2}\sqrt{\sum_{i=1}^{n} X2i^2}} \quad (1)$$

Where:- X1 and X2 are the subject of the email from the same recipient.

C. Entropy-Based

Entropy measures how diverse a value is in a given content. It is a concept mainly used in information theory.

- ***Entropy value between email subjects within the same sender***

We calculated the entropy value of the email subject within the sender. Since the data is aggregated using sender after calculating the entropy value between each, we consider the mean. For example, let X is the sequence of email subjects within the same sender, i.e., X = X1, X2, ..., Xn. the entropy between two subjects is calculated using eq 2

$$H(T) = \sum_{i=1}^{n} P(Ti)\ logP(Ti) \quad (2)$$

III APPROACH

The proposed system model is used to detect email spam by considering content-Based, Entropy-Based, Semantic-Based. The model can see and filter email as spam and non-spam. Figure1 shows the general approach that is followed in other similar studies as well. The main steps of the approach include data collection, Feature extraction, Preprocessing, model building (using Training data), Testing(using testing data), and performance evaluation. Activities under each phase are described in detail below.

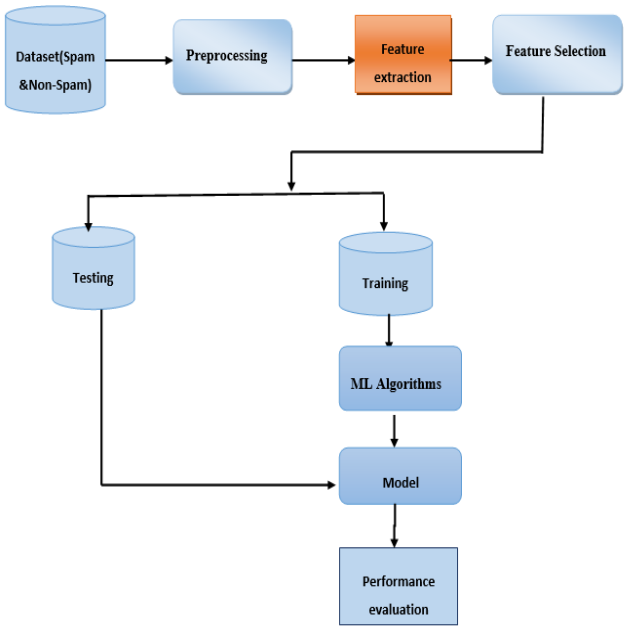


Figure 1: proposed system model for email spam detection.

The process in the attached methodology is described below.

A. Data Collection

The base for email spam detection is the collection of spam and non-spam data. The data is aggregated by sender per six hours. It also collects the existing features that previous researcher used in [6].

B. Preprocessing

For preprocessing we use Tokenization, Stop-word removal and stemming techniques.

- **Tokenization:-** It is used for preprocessing technique used to split the text to small pieces of words also used to remove punctuation[1].
- **Stemming :-**It reduces words to their root word[1].by using the porter stemming algorithm.
- **Stop word removal :-**Stop words are widely used words like stop words such as “the,” “a,” “an,” “in,” “is”. So, these techniques remove those words to give more emphasis on the vital information.

C. Model building

To build prediction models, we use four machine learning algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, Random Forest.

D. Evaluation

To assess the newly added features for email spam detection, we split the dataset into two, one for training and the other for testing. The training dataset is used to train the machine learning algorithm and build a model, while the test dataset is used for testing the performance of the model in predicting email. The performance of the model is evaluated using standard evaluation metrics.

IV EXPERIMENT

A. Dataset

For our experiment, we collected two months (February and March 2021) spam email and 21 days of non-spam email data from ethiotelecom. The total number of data collected
is shown in Table 1.

Table 4 Characteristic of the dataset.

Type	Spam	Non-Spam	Total
Total # of emails	4397	3701	8098
Total # of users	734	998	1732

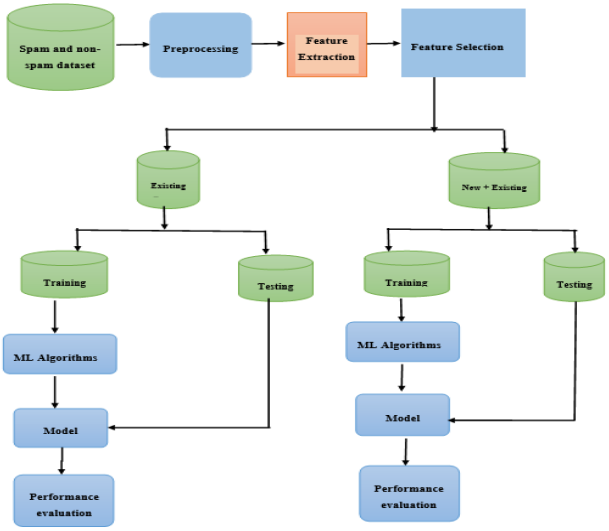
From the dataset, we extracted twelve features (six new and six existing). The extracted features and their data types are shown in Table 2.

Feature	Dataset
Number of recipients	Numeric
Number of emails sent	Numeric
Duration	Numeric
Character length	Numeric
Similarity	Numeric
Entropy	Numeric
Number of repetitive words	Numeric
Number of nouns	Numeric
Number of words	Numeric
Number of unique words	Numeric
Number of question marks	Numeric
Number of capitalized words	Numeric

TABLE 2: Characteristic of dataset.

B. Experiment setup

To evaluate the impact of the newly added features for improving the performance of email spam detection, we conducted two sets of experiments. In the first set of experiments, we used the content-based features prepared by Ammara Zamir et al. [6] to build and evaluate email spam detection models. In the second set of experiments, we used the new dataset to build and evaluate the email spam detection models. To avoid bias in selecting the training and testing dataset, we use 10-fold cross-validation.



C. Model building

The main objective of this paper is to improve the performance of the email spam detection technique. In this regard, we built four prediction models that map the given
input feature values to an output whose values are binary: spam and nonspam. The four prediction models are built using K Nearest Neighbor, Support Vector Machine, Logistic Regression, Random Forest. To build the models, we used the WEKA machine learning tool.

D. Evaluation Metrics

To evaluate the performance of our approach, we used widely used evaluation metrics, i.e., accuracy, FPR, precision, recall, F-measure, and ROC/AUC which are derived from a standard confusion matrix.

✓ **Accuracy**

$$\textbf{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

✓ FPR

$$FPR = \frac{FP}{FP+TN}$$

✓ **Precision**

$$\text{Precision} = \frac{TP}{TP + FP}$$

✓ **Recall**

$$\text{Recall} = \frac{TP}{TP + FN}$$

✓ **F-measure**

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

TABLE 3 Overall performance evaluation result.

Algorithm	Dataset	Accuracy(%)	Δ Accuracy	FPR(%)	AUC(%)	Precision(%)	Recall(%)	F1(%)	Δ F1
K Nearest Neighbor	Augmented	**93.19**	**5.49**	**6.5**	**93.9**	**91.3**	**92.8**	**92**	**6.6**
	Baseline	**87.7**		**10.2**	**93.3**	**86**	**84.9**	**85.4**	
Support Vector Machine	Augmented	**95.09**	**8.14**	**2.9**	**94.7**	**95.9**	**92.4**	**94.1**	**9.9**
	Baseline	**86.95**		**9.4**	**86.3**	**86.5**	**82**	**84.2**	
Logistic Regression	Augmented	**93.65**	**14.2**	**3.9**	**98.1**	**94.5**	**90.3**	**92.4**	**20.7**
	Baseline	**79.45**		**7.3**	**88.6**	**86.1**	**61.5**	**71.7**	
Random Forest	Augmented	**97.63**	**9.52**	**2**	**99.7**	**97.3**	**97.1**	**97.1**	**11.3**
	Baseline	**88.11**		**9.2**	**94**	**87.1**	**84.5**	**85.8**	

V. RESULTS AND DISCUSSIONS

The new+ existing (augmented)feature dataset F1-score for email spam detection is improved by 6.6%, 9.9%, 20.7%, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively. Tables I shows the results for the five machine learning algorithms. While using the new dataset, the F1-score of email spam detection is improved in all the algorithms.

The F1-score of the email spam detection is improved by 6.6%, 9.9%, 20.7%, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest, respectively. The most significant progress in the F1-score, 20.7%, is observed using the logistic regression algorithm. The F1-score of the existing feature dataset is lower than 90%in all the algorithms. Compared to the other algorithm, F1-score progress is relatively low in K Nearest Neighbor while using the existing feature.

The augmented feature dataset also shows improvement in accuracy for all algorithms. The improvements in accuracy are 5.49%,8.14%,14.2%, and 9.52% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest respectively. This shows that the newly added features help to improve email spam detection. The highest accuracy and F1-score are observed while using random forest followed by logistic regression. The lowest accuracy and F1-score are observed for the K Nearest Neighbor algorithm. This implies that machine learning algorithms in email spam detection also have a significant role in performance improvement.

The above result shows that the new feature helps in improving the performance of email spam detection. Among the five machine learning algorithms used in the study, Random Forest, followed by logistic regression and Support Vector Machine, gives the highest accuracy and F1-score measures while using existing and new datasets. K Nearest Neighbor has comparable low performance. The lowest accuracy and F1-score values are recorded for K Nearest Neighbor algorithm.

In this research question, we ranked the importance of each feature using the information gain and gain ratio algorithms (See Table 6.4). The result shows that the new features are ranked in the top six features (Table 4.8). This shows that the newly added features contribute to the improvement of email spam detection. While using the information gain algorithm, *length of address, the number of receivers, duration, sent emails, and entropy* is from the newly added features. The number of question marks is the least important feature.

Finally, the results show that the new six added features help improve email spam detection performance.

VI. RELATED WORKS

Several approaches are proposed to detect email spam. Based on the approach used for email spam detection, the approaches could be broadly classified as rule-based and machine learning-based approaches.

Rule-based email spam detection is one mechanism for email spam detection. By Setting some rules, if the incoming email fulfills that rule, that email is predicted as spam or non-spam. The approach is mainly applied to the client-side and server-side filtering techniques[3]. The main drawbacks of the rule-based spam filtering technique are if the incoming spam email does not fulfill the threshold they set, that spam email may be considered as non-spam and enter the user inbox.

Sultana et al.[1] used a machine learning approach to detect spam emails and identify the IP address of the spammer. The approach block all IP addresses of the spammer. Blocking the spammer IP, however, may cost more when there is a false negative. The proposed system uses a Naïve Bayes algorithm. The author uses the Kaggle dataset for training and testing the algorithm. However, the researcher does not use the sender information or other computed

features to improve the email spam detection technique. Nandhini, S et al.[2] conducts another literature that followed such an approach. Nandhini. S et al. compare the five machine learning algorithms using the UCI dataset. Using different performance matrices, they try to evaluate each algorithm. Ammara Zamir, et al.[6] uses feature centric concept, which means the authors use different feature categories like content-based, entropy-based, semantic-based, user-based, lexical-based features. In this research, it is seen that the more the features in the dataset, the better the accuracy. Regarding the content-based features, using different measurements may capture the spammer's characteristics from the content information. Similar to Ammara Zamir, et al.[6], Min Zhou et al.[14] a feature-centric approach for improving email spam detection. The researchers consider entropy-based spam detection. Using the entropy value of the subject and the content of the email to see if emails sent are diversified.

VII. CONCLUSION

Email spam detection approaches use the email feature for detecting email as spam or non-spam. In this paper, we propose the new email features with the existing features to improve the performance of email spam detection. The newly added features are extracted from the email header (from the sender and subject of the email). The newly added features are email sent, Receiver, Duration, length of the sender address, similarity within-subjects, and entropy value.

To assess the impact of the newly added features to improve email spam detection, we conducted two experiments using models built with five commonly used machine learning algorithms: K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest. The experiment compares the performance of the new + existing feature dataset with the existing features dataset. The results show that the new + existing dataset improves the performance of email spam detection in all the experiments. F1-score of the email spam detection is improved by 6.6, 9.9, 20.7, 11.3% while using K Nearest Neighbor, Support Vector Machine, Logistic Regression, and Random Forest, respectively, is observed on average 12.1% improvement using a new + existing dataset.

The results show that the new + existing dataset improves the performance of email spam detection. In this study, we used only six additional features. In the future , since more words appear in the email's body, we suggest using features extracted from the email body to further improve spam email detection in the future.

REFERENCES

[1] T. Sultana, “Email based Spam Detection,” *Int. J. Eng. Res.*, vol. V9, no. 06, pp. 135–139, 2020, doi: 10.17577/ijertv9is060087.

[2] S. Nandhini and D. J. Marseline, “Performance Evaluation of Machine Learning Algorithms for Email Spam Detection,” *Int. Conf. Emerg. Trends Inf. Technol. Eng. ic-ETITE 2020*, pp. 1–4, 2020, doi: 10.1109/ic-ETITE47903.2020.312.

[3] H. Bhuiyan, A. Ashiquzzaman, T. I. Juthi, S. Biswas, and J. Ara, “A Survey of Existing E-Mail Spam Filtering Methods Considering Machine Learning Techniques,” *Glob. J. Comput. Sci. Technol. Softw. Data Eng.*, vol. 1, no. 2, 2018.

[4] S. Suryawanshi, A. Goswami, and P. Patil, “Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers,” *Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019*, pp. 69–74, 2019, doi: 10.1109/IACC48062.2019.8971582.

[5] M. Zhiwei, M. M. Singh, and Z. F. Zaaba, “Email Spam Detection : a Method of Metaclassifiers Stacking,” *Int. Conf. Comput. Informatics*, no. 200, pp. 750–757, 2017.

[6] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal, and A. U. Akram, “A feature-centric spam email detection model using diverse supervised machine learning algorithms,” *Electron. Libr.*, vol. 38, no. 3, pp. 633–657, 2020, doi: 10.1108/EL-07-2019-0181.

[7] M. V. Madhavan, S. Pande, P. Umekar, T. Mahore, and D. Kalyankar, “Comparative analysis of detection of email spam with the aid of machine learning approaches,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, doi: 10.1088/1757-899X/1022/1/012113.

[8] N. Parmar, A. Sharma, and H. Jain, “Email Spam Detection using Naïve Bayes and Particle Swarm Optimization,” vol. 6, no. 10, pp. 367–373, 2020.

[9] D. M. Ablel-Rheem, A. O. Ibrahim, S. Kasim, A. A. Almazroi, and M. A. Ismail, “Hybrid feature selection and ensemble learning method for spam email classification,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1.4 Special Issue, pp. 217–223, 2020, doi: 10.30534/ijatcse/2020/3291.42020.

[10] R. Talaei Pashiri, Y. Rostami, and M. Mahrami, “Spam detection through feature selection using artificial neural network and sine–cosine algorithm,” *Math. Sci.*, vol. 14, no. 3, pp. 193–199, 2020, doi: 10.1007/s40096-020-00327-8.

[11] P. Sharma and U. Bhardwaj, “Machine learning based spam E-mail detection,” *Int. J. Intell. Eng. Syst.*, vol. 11, no. 3, pp. 1–10, 2018, doi: 10.22266/IJIES2018.0630.01.

[12] M. Vinodhini, D. Prithvi, and S. Balaji, "Spam Detection Framework using ML Algorithm," Int. J. Recent Technol. Eng., vol. 8, no. 6, pp. 5326–5329, 2020, doi: 10.35940/ijrte.f1120.038620.

[13] R. Talaei Pashiri, Y. Rostami, and M. Mahrami, "Spam detection through feature selection using artificial neural network and sine–cosine algorithm," Math. Sci., vol. 14, no. 3, pp. 193–199, 2020, doi: 10.1007/s40096-020-00327-8.

[14] M. Zhou, S. Zhang, Y. Qiu, H. Luo, and Z. Wu, "Entropy-based spammer detection," ACM Int. Conf. Proceeding Ser., 2018, doi:

10.1145/3240876.3240901.
[15] D. Mallampati and N. P. Hegde, "A Machine Learning Based Email Spam Classification Framework Model: Related Challenges and Issues," Int. J. Innov. Technol. Explor. Eng., vol. 9, no. 4, pp. 3137–3144, 2020, doi: 10.35940/ijitee.d1561.029420.
[16] L. Baoli, Y. Shiwen, and L. Qin, "An Improved k - Nearest Neighbor Algorithm," Proc. 20th Int. Conf. Comput. Process. Orient. Lang., no. July, 2003.