Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques

Mesay Shemsu Jemal

A Thesis Submitted to the Department of Computer Science and Engineering

College of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

February, 2025

Adama, Ethiopia

Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques

Mesay Shemsu Jemal

Advisor: Ketema Adere Gemeda (PhD)

A Thesis Submitted to the Department of Computer Science and Engineering

College of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

February, 2025

Adama, Ethiopia

Declaration

I, hereby declare that this Master thesis proposal entitled **"Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques"** is my own work and has not been submitted for the award of any academic degree, diploma, or certificate at any other university. All sources of materials that are used for this thesis have been duly acknowledged through citations.

Mesay Shemsu Jemal	_____	_____
Name of student	Signature	Date

Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled **"Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques"** prepared under my guidance by **Mesay Shemsu Jemal** submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering (CSE). Therefore, I recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Ketema Adere Gemeda (PhD)	_____	_____
Major Advisor	Signature	Date

Approval Page

I/we hereby certify that the recommendation and suggestion given by the proposal review committee are appropriately incorporated into the final thesis proposal entitled **"Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques"** by **Mesay Shemsu Jemal.**

Advisor/Supervisor	Signature	Date

We, the undersigned, members of the Board of Reviewers of the proposal open defense by **Mesay Shemsu Jemal** have read and evaluated the thesis proposal entitled **"Enhancing the Performance of Email Spam Detection and Classification using Machine Learning Techniques"** and assessed the understanding of the candidate about the proposed research. This is, therefore, to certify that the thesis proposal is accepted and we recommend the implementation of the proposal.

Chairperson	Signature	Date
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

Final approval and acceptance of the thesis/dissertation proposal is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

Acknowledgement

Primarily, I am deeply grateful to Almighty God for granting me the strength, wisdom, and perseverance to complete this research.

I would like to express my sincere appreciation to my advisor, Ketema Adere (PhD) for his invaluable guidance, encouragement, and constructive feedback throughout this journey. His expertise and support have been instrumental in shaping this work.

A heartfelt thank you to my family for their unwavering love, patience, and encouragement. Their constant support has been my greatest source of motivation.

Finally, I extend my gratitude to the Computer Science and Engineering Department staff members for their assistance, resources, and insightful discussions that have contributed significantly to the completion of this research.

Table of Contents

List of Figures

List of Tables

Acronyms

ML	Machine Learning
NLP	Natural Language Processing
AI	Artificial Intelligence
NB	Naïve Bayes
MNB	Multinomial Naïve Bayes
BNB	Bernoulli Naïve Bayes
SVM	Support Vector Machines
RF	Random Forest
LR	Logistic Regression
KNN	K Nearest Neighbor
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
DMNB	Discriminative Multinomial Naive Bayes
HAN	Hierarchical Attentional Hybrid Neural Networks
AUC	Area Under the Curve
URL	Uniform Resource Locator
PDF	Portable Document Format
TR	TREC
GS	GenSpam
SA	SpamAssassin
EN	Enron
LS	Ling Spam
WEKA	Waikato Environment for Knowledge Analysis

TF-IDF	Term Frequency-Inverse Document Frequency
CSV	Comma Separated Values
GLM	Generalized Linear Model
FLM	Factored Language Model
GBT	Gradient Boosting Trees
OOB	Out of Bag
CV	Cross Validation

Abstract

Email communication is regarded as the primary professional channel, enabling businesspeople and both commercial and nonprofit organizations to communicate with one another and exchange significant official documents and reports on a global scale. This global route draws a lot of attackers and intruders who utilize these innovations to commit crimes; in particular, the spammers create fake emails that contain attractive information and distribute them to clients worldwide. Current approaches primarily focus on widely spoken languages like English, resulting in a lack of comprehensive models tailored to underrepresented languages such as Amharic. This linguistic gap leads to poor performance in identifying spam emails in Amharic due to limited availability of preprocessed datasets and language-specific tools. Additionally, challenges such as imbalanced datasets, noise in translated data, and the absence of robust feature extraction techniques further exacerbate the problem. While methods like TF-IDF and ML algorithms, including SVM and RF, show promise, they still struggle with precision and recall trade-offs, especially for imbalanced classes. These limitations underscore the need for a systematic enhancement of data preprocessing, balanced dataset creation, and model optimization to bridge the gap and improve email spam detection and classification for Amharic and other low-resource languages. ML and NLP techniques used to identify spam. ML techniques frequently used in spam filtering to classify emails to either legitimate (ham) or spam (unsolicited messages). We studied the existing features and models to improve the accuracy, precision, recall and f-score. We created Amharic dataset by translating from local and publicly available English dataset using Google translate API. We split the data 70% for training and 30% to obtain result that ensure generalizability and we used 5 K-fold CV to ensure the model generalizability. We employed four ML models MNB, BNB, SVM, and RF to examine SVM model performs higher that other models by attaining 98.52% accuracy. We also compared the proposed results with baseline works and our proposed model performs better in detecting and classifying email spam data.

Keywords: Email spam, NLP, ML, Google Translate, TF-IDF, SVM, MNB, BNB, RF

Chapter One

1. Introduction

1.1 Background of the Study

The popularity of electronic communication has increased due to the Internet's broad use. Emails, or electronic messages, are among the most crucial electronic communication tools. Most people and businesses have at least one email account these days. The importance and widespread use of email increased by its instant message delivery, cost-free nature, and ease of use. (Dada et al., 2019)

Email may contains a number of vulnerabilities in addition to being useful. Email accounts compromises in a number of ways, such as when emails with adverts or other content infiltrate your computer and install malware on it. When you click on the advertisement, the installed software can occasionally overload your bandwidth, which might interfere with communication. (Keskin & Sevli, 2024)

The concept of spam predates the internet, with its roots in physical chain letters. The first recognized instance of email spam occurred in 1978 when Gary Thuerk, a marketing manager at Digital Equipment Corporation, sent an unsolicited mass email to 393 recipients on ARPANET (the precursor to the modern internet). This email advertised a new product and sparked significant backlash, marking the beginning of the battle against unsolicited emails. The term ‘Spam’ is not clear where the word comes from, but most authors wrote that the term taken from Monty Python’s sketch. (Sanz et al., 2008)

The 1990s as email became more accessible, it became an attractive medium for marketers. Notable incidents, such as the infamous 1994 “Green Card Lottery” spam, demonstrated the disruptive potential of spam. Businesses recognized the need for solutions to manage and mitigate spam during this period. In the early 2000s, significant advancements in spam filtering technology occurred. Bayesian filters, which used statistical methods to identify spam based on email content rather than just sender information, improved accuracy and reduced false positives. Blacklists, whitelists, and CAPTCHA systems also played a role in combating spam. Email spam has been a persistent problem since the mid-1990s when commercial use of the internet first became possible. In 2004, to combat the problem software and labor needed including productivity lose. (Sanz et al., 2008)

However, with this convenience comes the pervasive issue of email spam, also known as unsolicited bulk email. Spam emails are unsolicited and often contain advertisements, frauds, malware, or phishing attempts. Initially, spam primarily used as a tool for advertising various products. Over time, however, it evolved to encompass malicious activities such as phishing and the distribution of malware more. Journals have documented the evolution of spamming techniques and the tactics employed by spammers to bypass filters and security measures. According to (Rao & Reiley, 2012) in the Artificial Intelligence (AI) review journal, email spam costs businesses up to $20.5 billion each year.

Email spam detection and classification has been an active area of research for decades, driven by the ever-increasing volume of spam emails. Early approaches relied on simple rule-based filters, such as checking for specific keywords or sender addresses. However, these methods quickly became ineffective as spammers adapted their techniques. (Sonare et al., 2023)

With the advent of machine learning (ML), methods that are more sophisticated emerged. Researchers began exploring various ML algorithms, including Naive Bayes (NB), Support Vector Machines (SVMs), and neural networks, to classify emails as spam or ham (non-spam). (Ahmed et al., 2023) These methods showed significant improvements over rule-based approaches, achieving high accuracy rates in many cases. (Ghosh & Senthilrajan, 2023)

Organizations and researchers seek to create reliable spam email detection filters in order to protect users' security and integrity. Though users continue to report an increasing amount of fraud and threats via spam emails, most ML based spam filters show very good performance. This area has two primary difficulties: the dataset shift problem, which can arise in a highly dynamic setting, and the antagonistic entity known as spammers. Studies that concentrate on the issues that arise from this dynamic context differ from traditional examinations of spam emails. Researchers examine several spammer tactics for infecting emails and the most recent methods for creating ML based filters. (Jáñez-Martino et al., 2023)

The challenges presented by changing spam tactics, including phishing attempts, social engineering, and the sophistication of spam content, are the main focus of current research. (Sonare et al., 2023) In order to increase the precision and resilience of spam detection systems, researchers are also looking into the application of Natural Language Processing (NLP) techniques to better comprehend the context and meaning of email content. (Shamoo, 2024)

Overall, email spam detection and classification has changed a lot over time, and ML and deep learning have been essential in increasing efficiency and accuracy. In order to stay up with the changing threat landscape, researchers will have to create novel and inventive methods as spammers continue to adapt.

1.2 Motivation

The growing number of internet users in today's globe has made email spam a significant problem. Emails can be can be categorized as ham (legitimate) or spam. Email spam, sometimes referred to as junk mail or unsolicited emails, is a kind of email that can hurt users by robbing them of important data, wasting their time, and resources. This problem is growing, and more research needs to publish to enhance email spam detection and classification.

The ramifications of spam activities have been rising globally, affecting users all over the world. (Jáñez-Martino et al., 2023) Even though the majority of ML based spam filters that published in academic journals have recently shown high performance, users continue to report an increasing number of frauds and attacks via spam emails. Email has become a target for hackers because so many people use it for communication these days. Mails must divided into two categories spam and ham in order to stop these kinds of attacks.

The main challenges in this field are dataset shift and adversarial figure. The environment is highly dynamic and prone to the dataset shift issue. A change in the dataset could significantly impair the estimated generalization performance. Spammers use different strategies for contaminating emails such as using different language. While spam detection and classification extensively studied for major languages, there is a pressing need to extend this research to languages such as Amharic.

Amharic is one of official language in Ethiopia, spoken by millions of people. By developing effective spam filters for Amharic, we ensure that users in this linguistic community can enjoy a safer online experience. This problem can be tackle by using public and local datasets, up to date features and better ML models. The primary advantages of ML-based spam filters over traditional methods include adaptability, improved accuracy, and the ability to handle large volumes of data. They can adapt to new spam trends by continuously learning from new data, making them more resilient to the constantly evolving tactics of spammers.

1.3 Problem Statement

Email spam is a significant problem in the field of cybersecurity. Filtering email is an important research topic aims to identify and filter out unwanted or unsolicited messages from legitimate ones. Spam emails are still a concern on the Internet because multiple copies of the same messages or commercial advertisements found in spam emails. Several ML techniques developed to detect and classify email spam. According to (Siddique et al., 2021) study the research focuses on automated ways to detect spam emails written in Urdu language. The study used ML algorithms. However, in the current situation detecting and classifying spam emails are not effective enough to tackle the problem. It does not incorporate other languages such as Amharic text, to enhance spam email detection. (Assegie, 2023) identify several limitations and gaps in the field of email spam detection using ML algorithms. It highlights the varying performance of different ML models, with the Random Forest (RF) model showing the highest accuracy. However, the study acknowledges the need for further research to improve the precision of these models. It suggests that future work should focus on testing the models with larger datasets to enhance their practical utility in detecting spam emails effectively. The paper also notes the lack of comparison between existing ML models, emphasizing the importance of employing various performance measures beyond accuracy, such as precision, recall and F-score, to provide a more comprehensive evaluation of the models' capabilities. Using recent public and local dataset it is possible to create a new Amharic dataset using Google translator API. The task is to propose a ML model that classify each mail as spam or ham (legitimate) with better accuracy. The proposed model will enhance the existing email spam detection and classification using existing features considering Amharic language.

1.4 Research Questions

The purpose of this research is to respond to the following research questions.

RQ1: How the existing non-English dataset used for Email spam detection and classification?

RQ2: How a new Amharic dataset can be created using Google translator API?

RQ3: How ML model developed to enhance the performance of Email spam detection and classification?

RQ4: How to evaluate the performance of the proposed solution?

1.5 Objectives

1.5.1 General Objective

Detecting and classifying spam emails from legitimate emails using ML model with NLP techniques.

1.5.2 Specific Objectives

- To analyze the existing spam detection and classification techniques.
- To design the proposed architecture for enhanced email spam detection and classification.
- To preprocess the data using NLP techniques for better accuracy.
- To propose a better ML model for email spam detection and classification enhancement.
- To evaluate the utilized model using selected ML metrics.

1.6 Scope and Limitation of the Study

1.6.1 Scope of the Study

This study focuses on Enhancing Email Spam Detection and Classification using ML models. Within the limitations of time and money, the thesis study's scope includes the following topics.

- This thesis considered only to three Email Spam behaviors such as, Phishing, Lottery Scams, and Advance-Fee Scams.
- This thesis focus on detection and classification considering Amharic data only.

1.6.2 Limitation of the Study

- The Google Translate API generally limits the requests to 5,000 characters.
- This study does not consider all Email Spam attack behaviors and other languages except Amharic due to time and resource constraints.

1.7 Organizations of the Thesis

The content of the thesis developed in this way: The broad research topic, the inspiration behind it, the motivation, the problems, the objectives, the thesis contribution and the measures necessary to get there, all outlined in the first chapter. In chapter 2, the literature review and related works are covered. The techniques or methods covered in chapter 3. In chapter 4, we have discussed the proposed solution for enhancing email spam detection and classification and classification using ML. In chapter 5, we have discussed the steps taken for implementation of the proposed solution and in chapter 6 result and discussion section, we discuss the results obtained from each model experiment and the evaluation metrics used including the research question discussion. Finally, in chapter 7, we wrote our conclusion and the future works that helps the next researcher to extend this research.

Chapter Two

2. Literature Review

2.1 Overview of Email Spam Detection and Classification

Email spam is also known as unwanted junk email or unsolicited email. According to reports, email is actually the most cheapest, popular, and fastest means of communication these days. (Olatunji, 2019) The term "spam" originated from "Shoulder Pork HAM," a canned precooked pork that was first commercialized in 1937. Over time, digital mail scams have eventually appropriated the term. (Paswan et al., 2012) Email spam is a type of electronic mail spam where unsolicited messages sent in bulk by email. (Zhou et al., 2010) This phenomenon has grown exponentially with the rise of digital communication, posing significant challenges to individuals and organizations alike. Spammers spread spam emails for purely commercial gain, which can lead to more unethical behaviors including financial instability and harm to one's reputation on a personal and institutional level. Spam is currently becoming more and more common in other digital communication platforms. (Karim et al., 2019) Spam emails can waste users' time, computing resources, and steal valuable information. Email users lose countless hours every day. They also lose time removing those messages in addition to the time they waste reading spam. Reading and deleting these messages takes time, which reduces work productivity, regardless of how many or how few a person receives each day. (Sanz et al., 2008) According to Three-way decision approach to email spam filtering many classification techniques provides a systematic calculation of the threshold values using Bayesian decision theory to provide feedback to users for handling their emails to minimize error rate. (Zhou et al., 2010) According to four algorithms classification using WEKA library a data mining tool result shows BayesNet and J48 performs better than SVM, but the deviation shows the algorithms performance depends on the data. (Sharaff et al., 2016) Traditional classification techniques may be susceptible to "Concept Drift," which is why they were unproductive for the lifelong classification of spam emails, although producing good classification results. (Mohammad, 2024) Spam detection and filtration are significant and enormous problems for email service providers nowadays.

Spammers may use tactics such as email harvesters, which methodically scan the web for email addresses, and phishing schemes, which trick users into revealing personal information. Spam filters frequently employ the Bag-of-Words (BOW) as one of their representations (Douzi et al., 2020) Term Frequency-Inverse Document Frequency (TF-IDF) to quantify the importance of a term within a document relative to its occurrence across multiple documents. (Kaddoura et al., 2020) By using Roman Urdu, the authors (Afzal & Mehmood, 2016) collected 1463 tweets, of which 1038 were classified as spam and 425 as ham. Discriminative multinomial Naive Bayes (DMNB) algorithms applied to that data. They received 95.42% with NB and 95.12% with DMNBText. Without regard to language or domain specifics, the methods applied to a word sequence that was numerical in nature. The performance of classification predicted to enhance by linguistic approaches, such as those that consider the contextual features of significant terms in Roman Urdu literature. In (Mohamad & Selamat, 2015) the effectiveness of hybrid feature selection for email classification assessed. Out of the 169 emails they gathered, 114 were spam and 55 were ham. They were able to obtain the maximum accuracy rate by using hybrid feature selection. They used four reductions of Hybrid Feature Selection (TF-IDF) and attained an accuracy of 84.8%. The Malay language dataset did not provide the best level of accuracy for the intended study, and it was not appropriate with TD-IDF and rough set theory. This is the flaw of their work. One of the Spam emails with several file extensions attached, together with URLs packed with dangerous and spammers may send content that contain spam. As a result, the recipient may experience identity theft, financial fraud, or data breaches. This literature review examines the key studies, theories, and findings related to email spam, highlighting the evolution of spam tactics, detection methods, and preventive measures.

2.2 Email Overview

Email is an electronic communication tool available to everyone with internet access.(Suryawanshi et al., 2019) Through electronic means, the communication transmitted from one user to another. It enables message sending and receiving between two or more users. The number of people using email has increased significantly because of the rise in internet users. Email utilized for a variety of purposes, including business, education, and managing daily operations because it is economical and quick means of communication. There is an extra flow when sending and receiving messages over this communication medium. The stages

involved in sending and receiving email called mail flow steps, and they provide insight into how multiple email servers' services coordinate with one another. Some startling numbers represents the adoption of email as a communication tool. There were approximately 5.5 billion active email accounts as of 2017. In 2019, this figure expected to increase by more than 5.5 billion. By the start of 2019, about one-third of the world's population expected to be using email. Approximately 236 billion emails sent and received every day as of 2018, with 53.5% of those emails being spam. In actuality, 14.5 billion spam emails sent per day on average in 2018. The FBI has revealed that spam emails cost corporate email users USD 12.5 billions in losses in 2018. In a few years, the amount of money that businesses lose as a result of this spamming attack may explode, reaching an estimated total of USD 257 billion by the middle of 2020. An estimated USD 20.5 billions will be damaged annually. (Karim et al., 2019)

Table 2- 1: Financial loss incurred in Australian markets due to digital fraud.

(*Statista - The Statistics Portal for Market Data, Market Research and Market Studies*, n.d.)

Year	**Total Loss (AUD)**	**Losses in Top Three Categories (AUD)**
2018	107,025,301	• Investment Scams – 38,846,635 • Dating & Romance – 24,648,024 • False Billing (Fake Invoices) – 5,512,502
2017	90,928,622	• Investment Scams – 31,327,476 • Dating & Romance – 20,530,578 • Other Business & Employment – 5,270,948
2016	83,561,599	• Dating & Romance – 25,480,351 • Investment Scams – 23,631,338 • Advanced Pay & Fee Fraud – 6,499,604

According to Table 1's discussion, the tendency for digital theft is rising year, and the numbers above indicate that email spam will only increase as a result of the growing use of this media. Investment scams primarily target those seeking for romantic companions online, whereas dating scams prey on people who exchange a sizeable sum of money for fraudulent but tempting

business prospects. Emails remain the criminals' first choice when it comes to spreading malware to carry out these kinds of scams. Within the first two and a half months of 2019, Australian individuals and businesses have already lost up to AUD 56,000 as a result of email fraud, according to recent data.

2.3 Email Architecture

As seen in Figure 2-1, email is currently one of the most widely utilized Internet tools for communication. (Altulaihan et al., 2023) Its architecture is composed of many different parts. As seen in Figure 2-1, email is currently one of the most widely used Internet applications and utilized by the majority of people for communication. Its architecture is composed of many different parts. To clarify the email forensics tools and approaches, we go over the email service's design and operation. Email messages used to be brief and mostly comprised of text. Emails can now include text, video, and audio messages.

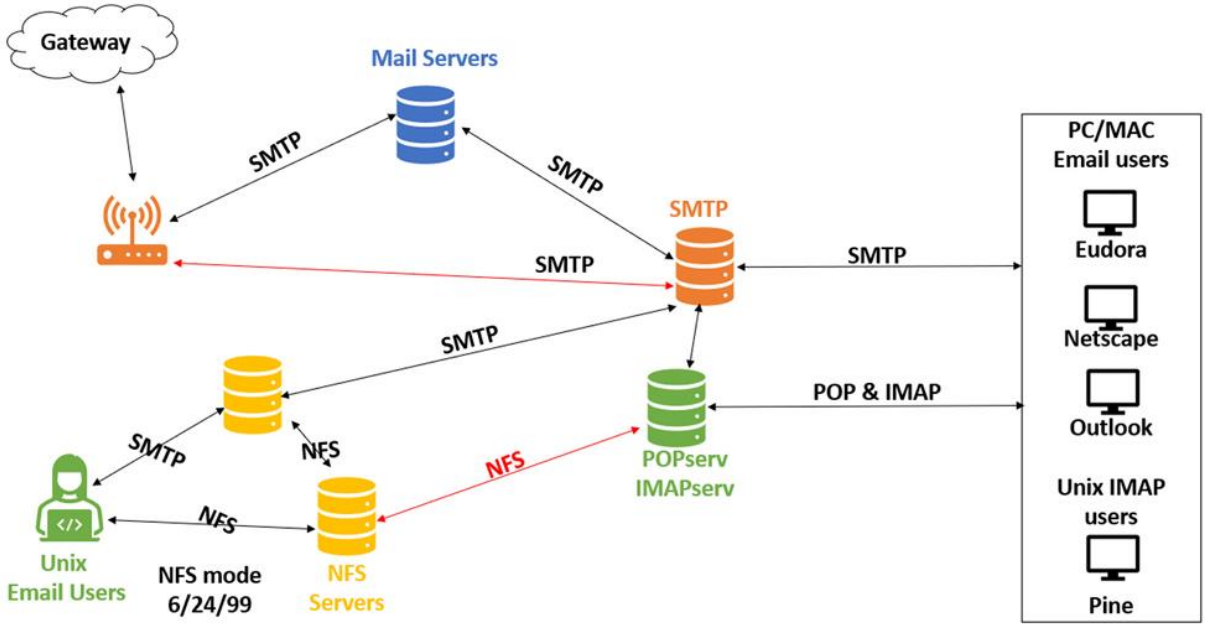


Figure 2-1: An Email System Architecture

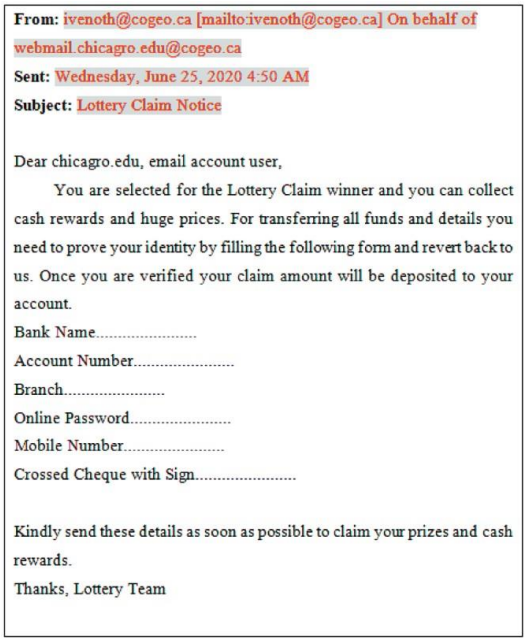
Figure Source Adapted from: (Altulaihan et al., 2023)

"User Agents," "Message Transfer Agents (MTA): Simple Mail Transfer Protocol (SMTP)," and "Mail Access Agent (MAA): Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP)" are the three primary components of the email system. Message composition,

reading, replying, forwarding, and mailbox management all done in user agents to facilitate message sending and receiving.

2.4 Email Spam

Spam defined as unsolicited bulk emails that sent to users without their consent. (Madhavan et al., 2021) Spreading pointless electronic material to victims with the intention of causing them financial and psychological harm is the main objective of spammers. Sending a large number of unsolicited emails to end users is how spammers get access to email servers. Numerous absurd emails from spam overload inboxes have the potential to harm the system in multiple ways. Spam slows down bandwidth, exposes users to dangerous content, crashes systems, and lowers worker productivity.



From: ivenoth@cogeo.ca [mailto:ivenoth@cogeo.ca] On behalf of webmail.chicagro.edu@cogeo.ca
Sent: Wednesday, June 25, 2020 4:50 AM
Subject: Lottery Claim Notice

Dear chicagro.edu, email account user,

You are selected for the Lottery Claim winner and you can collect cash rewards and huge prices. For transferring all funds and details you need to prove your identity by filling the following form and revert back to us. Once you are verified your claim amount will be deposited to your account.

Bank Name.....
Account Number.....
Branch.....
Online Password.....
Mobile Number.....
Crossed Cheque with Sign.....

Kindly send these details as soon as possible to claim your prizes and cash rewards.

Thanks, Lottery Team

Figure 2-2: Spam Email Sample

Source Adapted from: (Ismail et al., 2022)

2.5 Email Spam Attack Types

Email spam attacks come in various forms, each with distinct characteristics and objectives. Here are some common types:

- **Phishing**: Fraudulent emails designed to trick recipients into providing sensitive information like passwords, credit card numbers, or personal identification details. These emails often appear to be from legitimate sources.(Hong, 2012)
- **Spear Phishing**: A more targeted form of phishing where attackers tailor their messages to a specific individual or organization, using personal information to make the email more convincing.(Alavi et al., 2016)
- **Whaling**: Similar to spear phishing, but targeting high-profile individuals such as executives, CEOs, or other senior management within an organization.(Ferreira et al., 2015)
- **Spoofing** Emails that appear to come from a trusted source but are actually sent from an attacker. This often involves forging the sender's address to trick the recipient into thinking the email is legitimate.(Jakobsson, 2005)
- **Malware**: Emails that contain malicious attachments or links that, when opened, download malware onto the recipient's computer. Common malware types include viruses, ransomware, and spyware.(Egele et al., 2008)
- **Spam Bots**: Automated programs that send out large volumes of unsolicited emails, often promoting products, services, or phishing scams.(Ouyang et al., 2014)
- **Image Spam**: Spam emails that use images instead of text to bypass text-based spam filters. These images often contain malicious links or misleading information.(Fumera et al., 2006)
- **Snowshoeing**: Sending spam from multiple IP addresses and domains to evade detection by spreading out the volume of emails, making it harder for spam filters to block them all.(Balduzzi et al., 2010)
- **Pump-and-Dump**: A type of financial spam where emails promote a particular stock to artificially inflate its price, allowing the spammers to sell off their shares at a higher price before the market corrects itself.(Katsiampa, 2019)
- **Lottery Scams**: Emails claiming that the recipient has won a lottery or prize and requesting personal information or payment of fees to claim the prize.(Reyns et al., 2011)
- **Advance-Fee Scams**: These emails promise significant financial returns in exchange for an initial payment to cover fees or taxes.(Holt & Graves, 2007)

2.6 Advantages of Email Spam Detection

Nowadays, a company's email account is overflowing with unwanted correspondence. Spam detection is necessary to safeguard email users and secure businesses. Notifying the end user and the entire organizational network about spam emails in advance has several benefits. A few advantages of spam detection listed below.

Protect the data: Hackers can obtain corporate data through these spam emails and utilize it to harm the business. Emails that ask the receiver to click on a link or attachment, visit a website, or provide personal information that the recipient thinks is authentic. Once the data stolen, the spammers utilize it to access the user's information. Spammers send emails requesting money to deposit into their account or gain access to a user's personal or business network by tricking the receiver into opening an attachment. Consequently, email spam detection methods shield users from this kind of data loss. (Jáñez-Martino et al., 2023)

Protect the virus attacks: Spammers typically utilize spam emails to attack user applications, networks, and systems. They may cause any harm by using those spam emails. When we open an attachment or click a link in an email, many viruses, Trojan horses, and worms get triggered. Spammers may send messages with viruses. It will therefore be less likely for malware to attack if those emails found. (Sonare et al., 2023b)

Increase Productivity: If spam emails not filtered, they will show up in the inbox. They damage users' devices and take up a substantial amount of their time. As a result, users typically need to open practically all emails and determine if they include spam or not. Users may find it challenging to quickly recognize and reply to valid emails if there is an excessive volume of spam emails. Users would be able to save time and reduce the strain of sorting through legitimate emails with the aid of spam identification and filtering techniques. (Professor, 2021)

2.7 Machine Learning (ML)

These days, ML is a crucial component of numerous industrial, commercial, and scientific advancements. (Sarker, 2021) Data from the digital world, including the Internet of Things (IoT), cybersecurity, mobile, business, social media, health, and other sources, is abundant in

this Fourth Industrial Revolution era. Understanding AI, and more especially ML, is essential to creating intelligent assessments and automated applications. The discipline of ML has access to a wide variety of algorithms, including semi-supervised, supervised, unsupervised, and reinforcement learning techniques. Figure 1 shows the main algorithms under each category as well as the general organization of the many forms of ML.

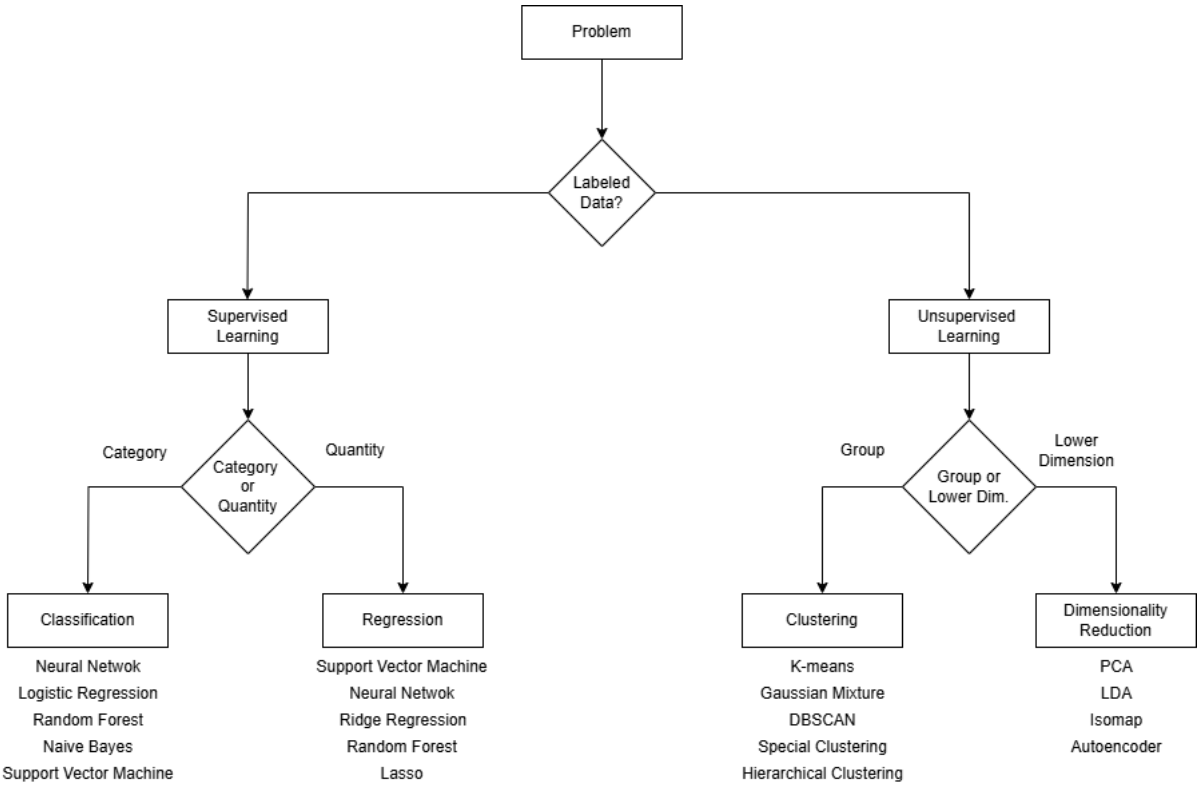


Figure 2-3: Types of ML

Random Forest Model (RF)

Maintaining the effectiveness and security of email communication networks depends on the detection of spam. Traditional rule-based filters are no longer sufficient to handle the growing number of spam emails, thus sophisticated ML approaches are required. One such technique that has gained popularity due to its accuracy and robustness in classification tasks is the RF approach.

The application of NB and RF classifiers to the identification of email spam examined. (Arya et al., 2023) Accuracy, precision, recall, and F1-score were among the criteria the researchers

used to assess the performance. According to the results, both classifiers did well, although RF showed promise in managing big feature spaces and preventing overfitting.

The RF and Gradient Boosting algorithms combined in this study's hybrid approach for email spam detection. The hybrid model uses both RF's ensemble learning capabilities and Gradient Boosting's boosting capability to increase classification accuracy. According on experimental data, the hybrid model performed better than the individual classifiers. (Sonare et al., 2023c)

Support Vector Machine Model (SVM)

In Email Spam Detection using SVM, the researchers evaluated the performance using metrics such as accuracy, precision, recall, and F1-score. The results indicated that SVM performed well in distinguishing between spam and non-spam emails, especially when combined with appropriate feature selection techniques. (Pandey et al., 2020)

This research focuses on the importance of feature selection in improving the performance of SVM classifiers. By selecting the most relevant features, the SVM model can achieve higher accuracy and reduce computational overhead. The study demonstrated that feature selection techniques could enhance the SVM's ability to detect spam emails effectively. (Verma, 2017)

The hybrid approach aims to exploit the strengths of both algorithms: SVM's ability to identify optimal hyperplanes and RF's robust prediction capabilities. The study showed that the hybrid model achieved higher accuracy, precision, recall, and F1-score compared to individual classifiers. (Bobde et al., 2023)

Naïve Bayes Model (NB)

The NB classifier, a probabilistic ML model, has been widely used for this email spam detection and classification due to its simplicity and effectiveness in text classification tasks. (Ezenwobodo & Samuel, 2022)

In the use of NB and RF classifiers for email spam detection the researchers explores and evaluated the performance using metrics such as accuracy, precision, recall, and F1-score. The results indicated that both classifiers performed well, with NB showing promising results in handling large feature spaces and mitigating overfitting. (Arya et al., 2023)

2.8 Natural Language Processing (NLP)

NLP is a subfield of AI focused on the interaction between computers and human (natural) languages. The goal of NLP is to enable computers to understand, interpret, and respond to human language in a valuable way. This encompasses a range of applications including language translation, sentiment analysis, speech recognition, and more.

This research on improving email detection and categorization must include Natural Language Processing (NLP) because NLP's capacity to comprehend and analyze human language is crucial for processing email content. By identifying patterns and features in textual data, natural language processing (NLP) techniques make it easier to distinguish between spam and legitimate emails. NLP-based techniques shown to be successful in spam detection. For instance, one study found that NLP is effective at processing text and identifying distinguishing characteristics of spam emails, resulting in improved detection accuracy. (Diyasa, 2023)
Here are the key areas in NLP:

Text Processing and Tokenization: Tokenization is the process of breaking down text into individual words or phrases, which is essential for most NLP tasks. This step allows for the analysis and processing of text data by transforming it into a more manageable format.

Part-of-Speech Tagging (POS): POS tagging involves assigning parts of speech to each word in a sentence (e.g., noun, verb, and adjective). This helps in understanding the grammatical structure and meaning of sentences.(Toutanova et al., 2003)

Named Entity Recognition (NER): NER is the process of identifying and classifying key entities in text (such as names of people, organizations, and locations). This is crucial for information extraction tasks.(Nadeau & Sekine, 2007)

Syntax and Parsing: Parsing involves analyzing the grammatical structure of sentences. Syntactic parsing can be used to understand the hierarchical structure of sentences, which is important for many NLP applications.(Collins, 2003)

Sentiment Analysis: Sentiment analysis involves determining the sentiment or emotion expressed in a piece of text. This is widely used in applications like social media monitoring and customer feedback analysis.(Pang & Lee, 2008)

Machine Translation: Machine translation is the task of automatically translating text from one language to another. This has seen significant advancements with the development of neural machine translation (NMT) models.(Vaswani et al., 2017)

Speech Recognition: Speech recognition involves converting spoken language into text. This technology is fundamental for voice-activated assistants and dictation software.(Hinton et al., 2012)

The field of NLP is rapidly evolving with advancements in deep learning and large-scale language models such as GPT-3 and BERT. These models have significantly improved the performance of NLP applications by leveraging vast amounts of data and computational power. (Devlin et al., 2018) NLP continues to be a dynamic and impactful field, driven by both theoretical advancements and practical applications. The integration of deep learning techniques has propelled NLP to new heights, enabling more sophisticated and accurate language processing capabilities.

There are a number of important benefits to using Natural Language Processing (NLP) in our thesis such as, automated literature reviews, qualitative data analysis, multilingual research and translation, text generalization and writing assistance, data mining from large text corpora, enhancing accessibility and inclusivity. (Yue Kang Zhao Cai & Liu, 2020)

2.9 Dataset Description

Email spam detection is a critical task in maintaining the efficiency and security of email communication systems. The success of email spam detection and classification relies heavily on the quality and diversity of datasets used for training and evaluating ML models. Researchers have used various datasets, ranging from publicly available collections to proprietary datasets, tailored to specific languages or spam types. This review examines the most common datasets

employed in this domain, highlighting their characteristics, limitations, and contributions to advancing spam detection research.

Spam Assassin Spam Corpus

This dataset is widely used in spam detection research. It contains a large collection of spam and non-spam emails, allowing researchers to train and test various ML algorithms. The dataset is known for its diversity and balance between spam and non-spam emails. (Wankhade et al., 2022)

Enron Email Dataset

The Enron Email dataset is another popular dataset used in spam detection research. It consists of emails from Enron Corporation, including spam and non-spam emails. The dataset is particularly useful for testing the performance of spam detection algorithms in a real-world corporate environment. The Enron spam dataset used in numerous studies to develop and evaluate spam detection techniques. (Metsis et al., 2006)

LingSpam Dataset

The LingSpam dataset consists messages collected from a linguistics mailing list, with a significant portion labeled as spam. It is smaller compared to other datasets but useful for academic research. (Androutsopoulos et al., 2000)

Custom Datasets

Researchers often compile their datasets from real-world email servers or scrape spam from public sources. Custom datasets tailored to meet specific research needs, such as handling highly imbalanced spam-to-ham ratios or testing novel spam tactics such as, phishing or malicious links. (Almeida et al., 2011)

2.10 Term Frequency – Inverse Document Frequency (TF-IDF)

For extraction TF-IDF techniques plays a crucial role in transforming the text data into numerical features that can effectively analyzed by ML models. TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (the corpus).

Term Frequency (TF): The first part of TF-IDF, Term Frequency (TF), measures how frequently a word appears in a single document. In the context of my thesis, TF captures the occurrence of specific words within individual emails. For example, if the word "offer" appears frequently in spam emails, the TF value for "offer" will be high in those documents. However, TF alone might not be enough, as common words across all documents would also have high TF values, leading to less informative features.

Inverse Document Frequency (IDF): addresses the limitation by assigning higher weights to words that are unique to specific documents. It does this by diminishing the weight of commonly occurring words across the entire dataset such as “እና”, “ነው”, “ላይ” in Amharic. The IDF calculated as the algorithm of the total number of documents divided by the number of documents containing the word.

2.11 Cross Validation

A crucial statistical method in ML, cross validation (CV) used to assess and improve the generalizability of predictive models. To ensure that every data point used for both training and validation, the dataset divided into several subsets. Models then iteratively trained on some of the data and validated on the remaining segments. This methodology offers a more reliable approximation of a model's performance on unseen data and helps mitigate overfitting.

K-fold, leave-one-out, and stratified K-fold are a few of the CV techniques that provide flexibility in model evaluation. Depending on the features of the dataset and the intended use, each technique has unique benefits and drawbacks. For example, leave-one-out CV is very useful for small datasets, even though it requires more computing power, whereas K-fold CV well known for its ability to balance bias and variance in performance estimation. Recent research has shown how the CV approach can significantly affect model performance and interpretation, highlighting the need for careful selection that is specific to the issue area. (Sweet et al., 2023)

Additionally, studies have shown that the choice of cross-validation methods can significantly affect ML models' performance, which in turn impacts the validity of model evaluation. (Kaliappan et al., 2023)

Therefore, creating strong and dependable machine learning models requires a thorough understanding of cross-validation techniques. An approach that frequently used in ML to evaluate a model's performance and generalizability is 5-fold CV. As part of this approach, the dataset divided into five equal subsets, or "folds." To ensure that each fold used as the validation set once, the model trained on four folds and validated on the remainder five times.

Why 5-fold CV?

Computational Efficiency: 5-fold CV uses less training rounds than higher-fold CV techniques, like 10-fold CV, which saves computing time and resources. This effectiveness is especially helpful when working with complicated models or big datasets. (Wilimitis & Walsh, 2023)

Reliable Performance Estimation: Although techniques such as leave-one-out CV rely on a single observation for validation, they might result in performance estimates that are very variable, particularly when dealing with small datasets. 5-fold cross-validation, on the other hand, provides a more consistent and trustworthy estimate of model performance by balancing bias and variance. (Refaeilzadeh et al., 2009)

2.12 Spammer Tricks

(Jáñez-Martino et al., 2023) stated that Spammers use various tactics to trick people into giving up their personal information or paying them. Spammers use various tactics to trick people into clicking on spam messages. Some of the most common types of spam messages include fake notifications from social networks, banking phishing, fake notifications from popular services and sellers, fake notifications from e-mail services, and poisoning text, obfuscated words, hidden text salting, and image-based spam. Fake notifications from social networks appear to come from popular social networks and are about new friends, activities, comments, likes, etc. Banking phishing messages sent in the name of banks or payment systems and related to account blocking or “suspicious activity” on the client’s personal account. Fake notifications from popular services and sellers created using the brand names of popular online stores,

delivery services, booking sites, multimedia platforms, job search websites, and other popular online services. Scammers use fake notifications from e-mail services to harvest usernames and passwords for e-mail services. Finally, poisoning text, obfuscated words, hidden text salting, and image-based spam are some of the most recent strategies used by spammers to trick people into clicking on spam messages. To tackle the spam email problem the following solutions can apply.

Spam email datasets: there is a need for more recent and diverse spam email datasets that reflect the current trends and challenges in this domain. Also, propose to unify the criteria for evaluating anti-spam filters using ML, taking into account the dataset shift problem.
Spammer strategies: the main spammer strategies used to evade spam filters, but they do not provide a comprehensive analysis of their impact on the performance and robustness of different classifiers. They also do not explore the possibility of detecting and anticipating new spammer tricks using data analysis or adversarial ML techniques.
Feature extraction and selection: Most researches focus on text-based feature extraction and selection methods, but they do not consider other types of features that could be useful for spam email detection, such as language base features. They also do not compare the effectiveness of different feature selection algorithms or criteria for this task.

To combat spam emails, ML models used to analyze the content of emails and identify patterns that are indicative of spam. Some popular ML models used for spam detection include NB, SVM, RF, and Logistic Regression. These models can help email service providers filter out spam emails and protect users from malware infections, data theft, and financial loss. (Jáñez-Martino et al., 2023) Nowadays, many ML algorithms used that show an accuracy over 90% to classify email data as spam or legitimate (ham). However, users are still reporting that there are still attacks that are spam emails by applying new strategies to bypass the spam filters.

In summary, email spam is a significant problem for service providers and users. Spammers use various tactics to send unsolicited emails that can waste users' time, computing resources, and steal valuable information. ML models used to combat spam emails and protect users from the harmful effects of spam emails.

2.13 Related Works

Research on Email spam detection and classification has significantly contributed to the cyber security field. Enhancing Email spam detection and classification is crucial research due to several reasons because Spammers continuously develop new techniques to bypass existing filters, such as using sophisticated language patterns, mimicking legitimate emails, or exploring new vulnerabilities. On the other hand, recent research often introduces new algorithms, methodologies, or hybrid approaches that improve the accuracy of spam detection systems. By incorporating the latest advancements, it is feasible to enhance the precision and reliability of classifying models, reducing false positive and false negatives. Reviewing related works can benchmark the proposed system against existing methods, identifying strengths and flaws. Keeping up with the latest research ensures the methods comply with current standards and regulations regarding data privacy and security. This is particularly important in handling email data, which often contains sensitive information. This research reviewed related journals, and articles. The paper reports the dataset source, methodology used and the accuracy results of their method.

According to (Siddique et al., 2021) study aims to detect emails written in Urdu using ML and deep learning (DL) models. The model Long Short-Term Memory (LSTM) achieved highest accuracy of 98.4% outperforming other models such as, NB, SVM, and Convolutional Neural Network (CNN) using a unique dataset of Urdu emails created and used for training and testing the models.

In (Ghogare & Dawoodi, 2023) research the proposed method outperforms the other methods in most of these metrics across five different datasets. For example, the proposed method achieves the highest AUC of 0.999 on the TREC (TR) 2007 dataset, compared to 0.994 for the second-best method. The paper also reports the results of cross-dataset evaluation, which shows that the proposed method is more robust and generalizable than the other methods. The paper used a content-based spam email classification that applies various text-preprocessing techniques like removal of stopwords, stemming, and lemmatization, and uses ML algorithms like NB, SVM, and RF to classify emails as ham or spam. The paper uses two standard datasets such as, Enron and SpamAssassin and one personal dataset (Yahoo mailbox) of email files in text format,

organized into spam and non-spam folders. The paper achieves impressive accuracy and F-score results on both standard and personal datasets, outperforming previous works. The preprocessing techniques enhance the classification performance by reducing noise and dimensionality. The result shows that RF algorithm proves to be the most suitable for spam email classification followed by SVM and NB.

The paper (Rastenis et al., 2021) addresses the challenge of classifying spam and phishing emails in multiple languages (English, Russian, and Lithuanian). This is a significant contribution as most existing research focuses on English-only datasets. The authors investigate the suitability of automated dataset translation for adapting email classification to different languages. This approach can be beneficial for organizations dealing with multilingual email traffic. The paper combines multiple datasets such as, Nazario, SpamAssassin, and VilniusTech then performs extensive preprocessing to create a balanced dataset for training and testing. This thorough preparation enhances the reliability of the results. The study uses datasets from specific sources. The generalizability of the results to other organizations or regions is not thoroughly tested.

(Assegie, 2023) compares the performance of different supervised learning models for email spam detection using accuracy, ROC-AUC, and F-score. The paper collected a dataset of emails from Kaggle, pre-processed and balanced the data using SMOTE, and trained eight supervised learning models on the data. The result shows that RF model achieved the highest accuracy (96.6%), and ROC-AUC (98.8%) among the models, while the XGBoost (XGB) model achieved the highest F-score (96.1%). The paper concluded that different models tend to perform differently in email spam detection, and suggested future work on testing the models on more data for real-time applications.

The study (Zavrak & Yilmaz, 2023) combines attention processes, Gated Recurrent Units (GRUs), and Convolutional Neural Networks (CNNs) to provide a unique method for email spam identification. Using a hierarchical attention technique that selectively focuses on significant portions of the email text to improve the model's spam detection capabilities is one of the paper's main accomplishments. To extract more significant, abstract, and generalizable information from the email text, the technique also mixes CNNs and GRUs. The study also includes cross-dataset evaluation, which aids in producing more autonomous performance

outcomes from the training dataset of the model. The suggested method uses temporal convolutions, which offer more adaptable receptive field sizes, to surpass current attention-based methods. Nevertheless, the study had a few limitations. Initially, the lack of a thorough comparison with other current approaches in the research makes it challenging to evaluate the suggested method's relative performance. Second, crucial elements for real-world application are the suggested method's memory utilization, computational cost, and training time, all of which are not disclosed by the authors. Finally, a critical component of real-world applications, the resilience of the suggested approach against adversarial attacks and noisy data is not assessed in the paper. Ultimately, even if the paper presents a promising method for detecting email spam, filling in these gaps could enhance its efficiency and applicability.

Table 2- 2: Related Works

Authors	Algorithms	Dataset Source	Max Accuracy	Advantages	Limitations
(Siddique et al., 2021)	LSTM, CNN, NB, SVM	Kaggle	98.4%	• The study focuses on detecting spam emails written in Urdu, a relatively unexplored area. • The authors created a unique dataset of Urdu, which is publicly available for future research.	• The study lacks relative comparison with other existing approaches.
(Rastenis et al., 2021)	NB, GLM, FLM, DT, RF,GBT, and SVM	SpamAssasin, Nazario, and VilniusTech	84%	• It addresses spam classification in multiple languages. • Detailed explanation for each steps.	• Accuracy could be improved by training datasets separately. • Lacks of comparison with other approaches.

(Ghogare & Dawoodi, 2023)	NB, SVM, and RF	Enron and SpamAssassin	99.2%	• The paper reports high accuracy and F-score results on all three datasets, outperforming previous methods.	• Lacks to measure the model to test how the model trained and fitted.
(Zavrak & Yilmaz, 2023)	HAN	TR, GS, SA, EN, LS	99.2%	Compares the results with state-of-the-art models and shows better results.	• Lack of comparison with other methods. • No report for the training time, computational cost or memory usage of the proposed method. • Lack of robustness evaluation.
(Assegie, 2023)	SVM, RF, KNN, DT, NB, LR, GB, and XGB	Kaggle	96.6%	It Compares the performance of eight supervised learning models.	• Limited performance measures. • The study used a single public dataset. • Lack of comparison with other approaches.

Chapter Three

3. Methodology

3.1 Chapter Overview

Users are more likely to trust and engage with email services that can effectively filter out spam and malicious emails in their native language. This trust is critical for maintaining secure and reliable communication channels. Enhancing email spam detection and classification using ML on a dataset containing Amharic emails is significant in many ways. Many existing spam filters primarily focus on English emails. By incorporating Amharic, the system can serve a wider audience, particularly in countries such as Ethiopia where Amharic spoken. Because multilingual capabilities ensure that the spam filter works regardless of the language of the email, they improve the accuracy of spam detection for users who get emails in both languages. Using Amharic dataset emails help increase the diversity of the training data, which can improve the accuracy and generalizability of the ML model. Cybercriminals often craft phishing emails in the recipient's native language to increase the likelihood of successful attacks. By training Amharic datasets, the detection system can recognize and block phishing attempts in Amharic language in addition to English, reducing the risk of users falling victim to these attacks.

Regarding the exploration of explanatory research inquiries, we make an effort to clarify the models or procedures we employ in the solution we propose. This chapter explains the entire process of Enhancing Email Spam Detection and Classification using ML methodology. This chapter explains Data Collection, Text Preprocessing, Tokenization, Vectorization, Feature Extraction and Selection, Model Selection and Training, and finally Model Evaluation.

3.2 Research Design

This study aims to improve the efficiency and accuracy of spam detection and email classification by leveraging ML techniques on Amharic datasets. By addressing the unique linguistic features and challenges inherent in processing emails in this language, this study seeks to develop a robust model that can seamlessly handle Amharic inputs, thereby contributing to the broader field of email filtering technology. This Amharic language approach not only broadens the applicability of findings but also provides valuable insights into the cross-

linguistic adaptability of ML models in the context of email security. Performance evaluation is another method used in this study to assess the model's robustness in terms of recall, accuracy, precision, and confusion matrix. In order to determine which aspects are most important and suitable for the target variable.

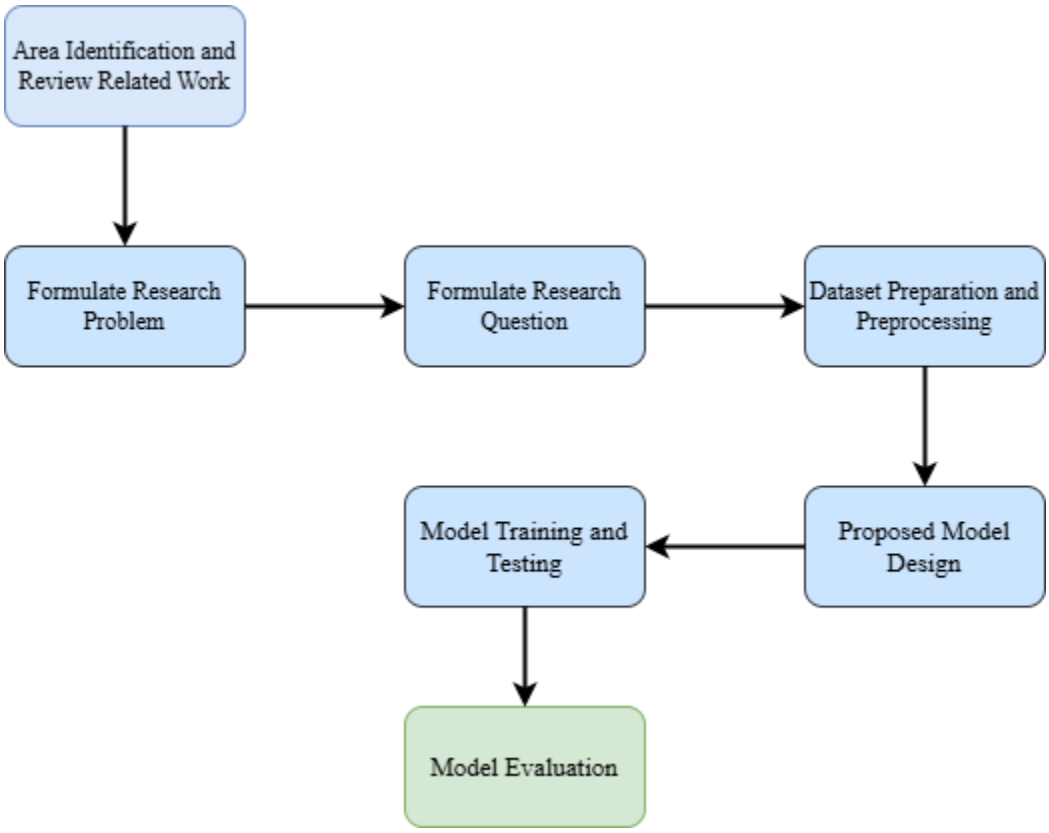


Figure 3- 1: Flow of Research Methodology

3.2.1 Area Identification

In our study, the first and most important phase in the research process is area identification. It entails determining a particular problem or knowledge gap that our research seeks to fill. The research question needs to be precise, manageable, and well defined.

3.2.2 Formulate Research Problem

Formulating a clear research question is crucial for several reasons. Firstly, it defines the scope of our investigation, helping us focus on specific aspects within a broader topic. Secondly, a well-crafted research question guides our study's objectives, ensuring that our

research remains on track and relevant. Additionally, it serves as a frame of reference for assessing our work, allowing us to evaluate whether we have addressed the central inquiry effectively.

3.2.3 Formulate Research Questions

A well-defined research question narrows down the scope of our investigation helping us to concentrate on specific aspects of our topic. These research questions acts as a compass guiding our study's objectives, ensuring that we stay on track and address the central inquiry effectively. When we formulate a clear research questions, we create a frame of reference that allows us to evaluate our work whether we have successfully answered the questions or not.

3.2.4 Dataset Preparation

A dataset is a structured collection of data, typically organized in a tabular form, where each row represents a unique record and each column represents a specific attribute or variable of the data. Datasets can vary in size and complexity, ranging from small and simple collections to large and complex ones. It used extensively in data analysis, ML, and research to train models, test hypotheses, and derive insights.

Datasets are essential in enabling researchers and analysts to systematically study and interpret data, facilitating the extraction of meaningful patterns and relationships. It can modify by translating from one language, such as English, into another language, such as Amharic. This process involves converting the textual data within the dataset into the target language while maintaining the original structure and meaning. Translation done manually by bilingual experts or automatically using machine translation tools and techniques.

However, Translations must done carefully to guarantee correctness and consistency because context can differ between languages. This process is particularly relevant in multilingual ML applications, where training data needs to be available in different languages to create models that can understand and process multiple languages effectively. We use the Email corpus, which has significant email spam features including legitimate emails, in our analysis. The Kaggle website provides public access to the dataset's source.

Table 3- 1 Email Dataset Sample

email	label
Upgrade to our premium plan for exclusive access to premium content and features.	0
Happy holidays from our team! Wishing you joy and prosperity this season.	0
We're hiring! Check out our career opportunities and join our dynamic team.	0
Your Amazon account has been locked. Click here to verify your account information.	1
Your opinion matters! Take our survey and help us enhance your experience.	0
Your payment has been received. Thank you for your prompt transaction.	0
Your email account storage is full. Click here to upgrade your account.	1
Dear [Name], thank you for subscribing to our newsletter. Here's your welcome gift!	0
Your account has been credited with loyalty points. Redeem them for exciting rewards!	0
You've been chosen for a free iPhone. Click here to claim your prize!	1
Don't miss out on our special offer! Sign up now and get a discount on your first purchase.	0
We're hiring interns for the summer. Apply now and gain valuable experience.	0
You're pre-approved for a loan. Click here to apply now!	1
We're thrilled to introduce our new collection. Shop now and enjoy exclusive discounts!	0
We're excited to announce our upcoming webinar series. Register now to reserve your spot!	0
We've added new features to our app based on your feedback. Update now!	0
You're a winner! Click here to claim your exclusive prize.	1
Your Facebook account has been hacked. Click here to secure your account.	1
Congratulations on reaching a new milestone! Here's to many more achievements.	0

3.2.5 Preprocessing

Data preprocessing is essential in preparing raw data for analysis by ML algorithms. In the context of email classification, this involves cleaning the data, handling missing values, and ensuring consistency across the dataset. The Kaggle dataset, while comprehensive, required significant preprocessing to transform it into a format suitable for the bilingual analysis central to our thesis. Given the nature of emails, which often contain noise such as metadata, HTML tags, and signatures, the preprocessing phase was crucial in filtering out irrelevant information and retaining only the meaningful text for analysis.

In the rapidly evolving field of ML, preprocessing data is a critical step that lays the foundation for accurate and effective model training. For our thesis, which focuses on enhancing email detection and classification, we utilized a publicly available email corpus obtained from Kaggle.com. This dataset provided a rich source of information that included various email components such as headers, bodies, and metadata. However, given the specific requirements of our thesis, which involved working with both Amharic languages. We decided to focus on translating the email headers particularly the subject lines into Amharic. This approach allowed me to overcome some of the practical limitations posed by the available translation tools while ensuring the quality and relevance of the data used in model training.

One of the key challenges we encountered during the preprocessing phase was translating the dataset's content into Amharic. Amharic, being a Semitic language with a unique script, posed additional challenges compared to more commonly spoken languages. Our initial plan was to translate entire email bodies to create a comprehensive bilingual dataset. However, this approach quickly proved impractical due to the limitations of the Googletrans library in Python, which restricts the amount of text that can be translate up to 5,000 characters per request.

To navigate this limitation, we made a strategic decision to focus on translating the email headers, specifically the subject lines. The subject line of an email is often a concise summary of its content, making it an ideal candidate for translation within the constraints of the Googletrans library. By translating the subject lines instead of the full email bodies, we able to maintain a manageable text size for each translation request while still capturing the essence of the email's content. This approach not only aligned with the library's character limit but also provided a meaningful dataset for training the classification model.

By focusing on translating email headers rather than entire bodies, we able to create a bilingual dataset that was manageable, however we used Amharic datasets only. The translated subject lines provided a useful linguistic bridge between the two languages, enabling the model to learn from Amharic data. This approach has significant implications for the broader goal of email detection and classification, particularly in regions where Amharic is widely spoken. The successful translation and preprocessing of email headers demonstrate that even with the

constraints of available tools, it is possible to create effective bilingual datasets for machine learning.

The key lies in strategic decision-making during the preprocessing phase, such as focusing on the most informative parts of the data and ensuring that the translation process aligns with the technical limitations of the tools used. We employed a strategy of splitting the CSV dataset into smaller, manageable chunks, each containing a portion of the data that fell within the character limit. By processing these smaller chunks individually, we successfully translate the data while adhering to the constraints of the translation tool. This approach not only ensured the completeness of the translation process but also maintained the integrity and accuracy of the

Table 3- 2 Amharic Email Dataset Sample

email	label
ለፕሪሚየም ይዘት እና ባህርያት ልዩ መዳረሻ ወደ ፕሪምየም አቅዳችን አሻሽል።	0
መልካም በዓል ከቡድናችን! በዚህ ወቅት ድስታን እና ብልጽግናን እመኛለሁ።	0
እየቀጠርን ነው! የስራ እድሎቻችንን ይመልከቱ እና ተለዋዋጭ ቡድናችንን ይቀላቀሉ።	0
የአማዞን መለያህ ተቆልፏል። የመለያዎን መረጃ ለማረጋገጥ እዚህ ጠቅ ያድርጉ።	1
የእርስዎ አስተያየት አስፈላጊ ነው! የዳሰሳ ጥናታችንን ይውሰዱ እና ልምድዎን እንድናሻሽል ያግዙን።	0
ክፍያዎ ደርሷል። ለፈጣን ግብይትዎ እናመሰግናለን።	0
የኢሜይል መለያህ ማከማቻ ሙሉ ነው። መለያህን ለማሻሻል እዚህ ጠቅ ኡድርግ።	1
ውድ [ስም]፣ ለጋዜጣችን ደንበኝነት ስለተመዘገቡ እናመሰግናለን። የእንኳን ደህና መጣችሁ ስጦታ ይኸውና!	0
መለያዎ በታማኝነት ነጥቦች ተቆጥሯል። ለአስደናቂ ሽልማቶች	0
ለነጻ iPhone ተመርጠዋል። ሽልማትዎን ለማግኘት እዚህ ጠቅ ያድርጉ!	1
የእኛን ልዩ ቅናሽ እንዳያመልጥዎ! አሁን ይመዝገቡ እና በመጀመሪያ ግዢዎ ላይ ቅናሽ ያግኙ።	0
ለበጋው ተለማማጆችን እየቀጠርን ነው። አሁን ያመልክቱ እና ጠቃሚ ተሞክሮ ያግኙ።	0
ለብድር አስቀድመው ተፈቅደዋል። አሁን ለማመልከት እዚህ ጠቅ ያድርጉ!	1
አዲሱን ስብሰባችንን ስናስተዋውቅ በጣም ደስ ብሎናል። አሁን ይግዙ እና ልዩ ቅናሾችን ይደሰቱ!	0
መጭውን የዌቢናር ተከታታዮቻችንን ለማሳወቅ ጓጉተናል። ቦታዎን ለማስያዝ አሁን ይመዝገቡ!	0
በአስተያየትዎ መሰረት አዳዲስ ባህርያትን ወደ መተግበሪያችን አክለናል። አሁን አዘምን!	0
እርስዎ አሸናፊ ነዎት! ልዩ ሽልማትዎን ለማግኘት እዚህ ጠቅ ያድርጉ።	1
የፌስቡክ አካውንትህ ተጠልፏል። መለያህን ለመጠበቅ እዚህ ጠቅ ኡድርግ።	1
አዲስ ምዕራፍ ላይ ስለደረስዎ እንኳን ደስ አለዎት! ለብዙ ተጨማሪ ስኬቶች እነሆ።	0

3.2.6 Design Proposed Solution

Designing a proposed solution is an essential step in our research process. It involves developing a solution to address the identified research problem and evaluating its feasibility. In Figure 2, we provided our design proposed solution.

3.3 Dataset Description

The quality and characteristics of the dataset used in a ML project are fundamental to the success of the models developed. In this research, which focuses on enhancing email spam detection and classification using ML, the dataset played a pivotal role in training, validating, and testing the models. The dataset used in this research was a collection of emails translated from English to Amharic, focusing on both spam and legitimate (ham) emails. The dataset sourced from publicly available email datasets and local dataset, which commonly used for spam detection research. The emails were initially in English and other foreign languages, translated to Amharic performed using the Google Translate API that used to explore the performance of ML models on non-English text. The dataset comprised total of 7053 records which is 1554 records from local (Ethio Telecom) and 5499 from public dataset, with each record representing an individual email. The dataset balanced, containing a mix of spam and ham emails to ensure that the models trained and tested on a diverse range of content. Each email in the dataset labeled as either (0) as "spam" or (1) as "ham," forming the basis for the supervised learning models developed in the research.

3.4 Dataset Preprocessing Techniques

Data preprocessing is crucial for our research, as it directly affects the accuracy and efficiency of the models we develop. Raw data, especially in the context of emails, often contains noise, inconsistencies, missing values, and irrelevant information, which can hinder the performance of ML algorithms. By cleaning, normalizing, and transforming the data during preprocessing, we ensure that the models receive high-quality input, leading to better pattern recognition and classifications that are more reliable. Preprocessing steps such as tokenization, stemming, and removing stopwords are particularly important in NLP tasks like email classification, as they help in reducing dimensionality and focusing on the most informative features. Moreover, when dealing with bilingual datasets, as in our case with Amharic emails, preprocessing helps in standardizing the data across languages, ensuring that the model can effectively learn from new language. Overall, effective data preprocessing sets the foundation for building robust, generalizable models that can achieve higher accuracy in detecting and classifying emails, which is essential for the success of our research.

3.4.1 Data Translation

Data translation known in the research area when the dataset lack happens. (Siddique et al., 2021) (Rastenis et al., 2021) In *Appendix 3* to create a Python program that reads a CSV file containing English text that translates it to Amharic, and then saves the translated content back to a new CSV file. We used the *pandas* library for handling the CSV files and a translation library like *googletrans* for translating the text. To handle text longer than this limit, we need to split the text into smaller chunks and translate each chunk separately. We used foreign language dataset that we obtained from public dataset source (Kaggle) and local source (Ethio Telecom Company) created Amharic dataset by splitting the datasets into chunks due to the *googletrans* API limitation. Then we merged the data into a single data frame and manually we correct the translated data to fix some translation errors.

3.4.2 Data Cleaning

Data cleaning in ML is the process of preparing raw data for analysis by addressing issues like missing values, duplicates, noise, and inconsistencies. This step is critical because poor-quality data can lead to inaccurate models and unreliable predictions. Common techniques include handling missing data through imputation or removal, correcting errors, normalizing or scaling features, and removing outliers. Effective data cleaning ensures that the dataset is consistent, accurate, and complete, which ultimately enhances the performance and accuracy of machine learning models. By thoroughly cleaning the data, we ensure that the input to the machine learning models is reliable and consistent, which is crucial for achieving accurate results in detecting and classifying emails across different languages. This step not only improves the model's accuracy but also contributes to the robustness and generalizability of the findings of the research. We employed data cleaning methods tailored to the specific characteristics of the Amharic datasets. Recognizing that each language presents unique challenges, such as different text structures, encoding issues, and language-specific noise. We carefully selected and applied distinct cleaning techniques to address these challenges effectively.

Data Cleaning for Amharic Data

The data cleaning process for an Amharic email dataset involves several specialized steps to address the unique characteristics of the language while preparing the data for ML.

Removing Duplicates and Irrelevant Content: the first step is to remove duplicate emails and irrelevant content such as headers, footers, and signatures. This ensures that the dataset consists of unique and meaningful data.

Handling Missing Values: Missing data in fields like subject lines or email body text can occur in any dataset. For an Amharic email dataset, it is important to either impute missing values using contextually appropriate methods or remove the records if the missing data significantly impacts the analysis.

Text Normalization: Amharic text normalization involves converting the text to a standard form. Since Amharic uses the Ge'ez script, normalization may include converting characters to a consistent encoding format (e.g., UTF-8) to avoid issues with different font representations. This step ensures that all text is in a uniform format, which is crucial for further processing.

Removing Stopwords: Stopwords are common words that do not contribute significantly to the analysis, such as "እና" (and), "ነው" (is), and "በ" (in). The stopwords list for Amharic obtained from repositories on GitHub. These stopwords removed from the dataset to reduce noise and focus on the more informative words in the text.

Tokenization: Tokenization for Amharic involves splitting the text into individual words or tokens. Given that Amharic is a morphologically rich language, tokenization might require specialized tools or custom algorithms accurately split words, considering the language's complex grammar and structure.

Removing Special Characters and Punctuations: Special characters, punctuation, and numbers within the Amharic text typically removed, as they may not contribute meaningfully to the classification task. Regular expressions used to clean the text by filtering out these elements, leaving behind only the relevant Amharic script.

Handling Language-Specific Challenges: Amharic, being a Semitic language, has unique linguistic features, such as rich morphology and complex verb conjugations. These characteristics require careful handling during data cleaning to avoid losing important contextual information. Customizing the data might be necessary to address these challenges effectively.

3.4.3 Feature Extraction

By allocating weights to terms according to how frequently they occur in a document compared to how frequently they occur in all documents, TF-IDF efficiently highlights terms that are important in particular texts while reducing the impact of common terms. This method improves the model's capacity to differentiate between various categories, including spam and genuine emails. (Qaiser & Ali, 2018)

Research has shown that in text classification tasks, TF-IDF frequently performs better than alternative feature extraction techniques. For example, studies comparing TF-IDF and Word2Vec models for emotion text detection discovered that TF-IDF modeling performed better. (Cahyani & Patasik, 2021)

In the context of dataset, which includes Amharic emails, TF-IDF applied separately to each datasets preprocessed text data. This results in two distinct feature sets that capture the unique characteristics of words in both datasets. These features then combined or processed individually, depending on the modeling approach, to create a robust feature representation that enhances the model's ability to classify emails accurately. By using TF-IDF, our research ensures that the ML models trained on features that highlight the most relevant words for classification while downplaying less informative terms. This leads improved performance in detecting and classifying spam versus legitimate emails in Amharic language, making the classification system more effective and reliable.

3.5 Parameter Setting

Parameter setting in ML is a critical step that involves selecting and tuning the hyper-parameters of a model to optimize its performance on a given task. In our research, we carefully give attention to the parameter settings to ensure that the models are well suited for the Amharic datasets. For the proposed solution, the learning rate fine-tuned through CV, ensuring that the model learns efficiently without compromising accuracy.

3.6 Model Training

For the proposed solution, we employed RF, SVM, and NB models to analyze and classify emails in Amharic. Each of these models offers distinct advantages, making them suitable for different aspects of the classification task.

- **Random Forest (RF)**

One of the traditional ML classifiers employed in other research is RF. According to (Sheneamer, 2021), RF outperforms other traditional classifiers in terms of accuracy, precision, recall, and F-measure. In our research RF model trained using the TF-IDF features extracted from the email dataset. The model creates several decision trees, each trained on a random subset of the data and features. This randomization helps in reducing overfitting and improving generalization. For each decision tree, the model selects a random subset of features to consider when splitting nodes. This feature selection ensures that no single feature dominates the prediction, which is particularly important in text data where certain words might be overly influential. RF was particularly effective in capturing complex patterns in the bilingual email dataset. Its ability to handle a mix of strong and weak features made it a reliable model for distinguishing between spam and legitimate emails across both languages.

- **Support Vector Machine (SVM)**

SVM chosen for the study because it is a supervised ML technique that performs well with categorized datasets. SVM is well known for its reliability as well as speed when working with small amounts of labeled data, and it is frequently employed for both classification and regression problems. According to (Siddique et al., 2021) the SVM model is a very successful classification technique because it employs a hyperplane to distinguish between positive and negative values (in this case, spam and ham emails). For the purpose of identifying and classifying email content, they emphasize SVM's high-precision classification capabilities.

The SVM model trained using the TF-IDF features, where the goal was to find the hyperplane that maximizes the margin between the classes (spam and not spam). In the context of email classification, SVM aims to find the decision boundary that best separates spam emails from legitimate ones. SVM is effective in high-dimensional spaces and performs well even when the number of features exceeds the number of samples, a common scenario in text classification. It is particularly strong in cases where the classes well separated in the feature space, which is beneficial when distinguishing between spam and non-spam emails. SVM’s ability to create complex decision boundaries made it a strong candidate for our research, especially for

handling the nuanced differences in language structure between Amharic and English. Its use of different kernels allowed the model to adapt to the specific characteristics of each language.

- **Naïve Bayes (NB)**

The NB model trained using the TF-IDF features, where it calculated the probability of an email being spam based on the individual probabilities of the words in the email. The assumption is that the presence of a particular word in a document is independent of the presence of any other word. For the thesis, both Multinomial Naive Bayes (MNB) and Bernoulli Naive Bayes (BNB) considered, with the final choice depending on which variant better handled the TF-IDF feature distribution.

We utilized MNB because it’s of suitability of data. MNB simulates the word distribution in documents; it works especially well for text categorization tasks. Because of this feature, it can effectively differentiate between spam and authentic emails. (Sadia et al., 2023)

We utilized BNB also because when the emphasis is on the appearance of particular keywords suggestive of spam, BNB's operation on binary term occurrence (the presence or absence of words) is advantageous. (De Luna et al., 2023)

The Multinomial variant typically used for text classification, as it works well with word frequency features. These models compared based on their performance metrics, such as accuracy, precision, recall, and F1-score. Each model's strengths leveraged, enhancing the overall classification system. The diversity of models allowed for a comprehensive analysis and robust classification of emails, improving the detection of spam across different languages. Overall, the use of RF, SVM, and NB provided a thorough exploration of ML techniques in the context of email detection and classification, demonstrating the effectiveness of combining different approaches to handle the complexities of a bilingual dataset.

3.7 Performance Evaluation Methods

A crucial instrument for determining the efficacy and dependability of the created model is performance evaluation. It is the procedure for calculating the precision and effectiveness of model detection on the necessary datasets. The major objective is to obtain important insights into the model's decision-making processes and learning process when executing the target problem's solution. The insight is used to evaluate many models, choose the optimal model for

the necessary tasks, and increase the accuracy and efficiency of model performance. The ML model uses a wide range of metrics, including regression and classification, but how each is applied varies depending on the task at hand. Here, we take into account classification metrics.

3.7.1 Confusion Matrix

A **confusion matrix** is a table used to evaluate the performance of a classification model. It provides a summary of prediction results by comparing the predicted class labels with the true class labels. It represented as a square matrix, with rows denoting the instances' actual class and columns their anticipated class. A matrix of 2x2 confusion reports the counts of TP, TN, FP, and FN for binary classification tasks, as indicated in the *Table 3-1*.

Table 3- 3: Conceptual Confusion Matrix

	Class "0"	**Class "1"**
Class "0"	**TN**	**FN**
Class "1"	**FP**	**TP**

- **True Positives (TP)**: The number of spam emails correctly classified as spam.
- **True Negatives (TN)**: The number of non-spam emails correctly classified as non-spam.
- **False Positives (FP)**: The number of non-spam emails incorrectly classified as spam (Type I error).
- **False Negatives (FN)**: The number of spam emails incorrectly classified as non-spam (Type II error).

3.7.2 Accuracy

Accuracy defined as the ratio of correctly predicted instances (both true positives and true negatives) to the total instances in the dataset. In binary classification, this metric indicates how well the model distinguishes between spam and non-spam emails.

$$Accuracy = \frac{Number\ of\ correct\ prediction\ (TP+TN)}{Total\ number\ of\ predictions\ (TP+TN+FP+FN)} \qquad (1)$$

3.7.3 Precision

Precision is the ratio of true positive predictions (correctly identified spam emails) to the total number of positive predictions (both true positives and false positives). In simpler terms, it measures how many of the emails predicted as spam are actually spam. High precision indicates that the model has a low rate of false positives, meaning it is effective at not labeling non-spam emails as spam. This is particularly important in email classification to avoid incorrectly categorizing important emails as spam.

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

3.7.4 Recall

Recall (also known as sensitivity or true positive rate) is the ratio of true positive predictions (correctly identified spam emails) to the total number of actual positives (the sum of true positives and false negatives). In other words, recall measures the ability of the model to detect all spam emails present in the dataset. High recall indicates that the model successfully identifies most of the spam emails, minimizing the number of spam emails that go undetected (false negatives).

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

3.7.5 F1-Score

The **F1-score** is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall. It is particularly useful when you need to ensure that both false positives (incorrectly labeling non-spam as spam) and false negatives (failing to identify actual spam) minimized.

$$F1Score = \frac{TP}{TP+FP} \qquad (4)$$

3.7.6 False Positive Rate (FPR)

The **False Positive Rate (FPR)** is the ratio of false positives to the sum of false positives and true negatives. The FPR measures the proportion of non-spam emails that incorrectly labeled as spam by the model.

$$FPR = \frac{FP}{FP+TN} \qquad (5)$$

3.7.7 False Negative Rate (FNR)

The **False Negative Rate (FNR)** is the ratio of false negatives to the sum of false negatives and true positives. The FNR measures the proportion of actual spam emails that not identified by the model, which reflects the model's ability to detect spam.

$$FNR = \frac{FN}{FN+TP} \qquad (6)$$

3.8 Design and Development Tools

Tools used for Design

This covers the hardware, software, and other platforms that utilized for file storage, processing, referencing, writing and testing code, and putting the suggested solution into practice.

- **Software Tools**

Jupyter Notebook: Since Python is the most widely used programming language for research, ML, data visualization, and data analysis, Jupyter Notebooks support a wide range of programming languages. In the notebook, users can interactively enter and execute code in the cells, with instantaneous results displayed.

Python: is a highly favored programming language for ML due to its simplicity, readability, and extensive ecosystem of libraries and frameworks. Its clear syntax allows researchers and practitioners to implement ML algorithms with ease and maintain readability and manageability in their code. Libraries such as Scikit-learn offer a broad range of tools for data preprocessing, model training, and evaluation, making Python an excellent choice for both beginners and experts. The language's versatility extends to its compatibility with data manipulation and

analysis libraries like Pandas and NumPy, which are essential for handling large datasets and performing numerical computations efficiently. Visualization libraries such as Matplotlib and Seaborn enhance Python's capability to present data insights and model performance metrics effectively.

Mendeley: is a reference management and academic social networking software designed to assist researchers in organizing, managing, and sharing their research materials. It allows users to import, organize, and manage academic papers and references. It supports a wide range of citation styles and helps users create and maintain a personal library of research materials.
Microsoft Office: we used Microsoft Office packages such as, Word, Excel, and PowerPoint for writing documentation, dataset visualization, and presentation.

- **Hardware Tools**

We used hardware tools shown in Table for this study.

Table 3- 4: Hardware Tools

Tool Name	Description
Processor	Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.19 GHz
Installed Memory (RAM)	8.00 GB
Hard Drive	1TB SSD

Chapter Four

4. Proposed Solution for Email Spam Detection and Classification Using ML

4.1 Chapter Overview

In this chapter, we offered a remedy to the current issue. The primary goal of the project is to improve email spam detection and classification performance by utilizing ML models including RF, SVM, MNB, and BNB. The model's performance was evaluated using metrics such as accuracy, precision, recall, F1 score, and confusion matrix, which were determined to be crucial components of any research activity. An enhanced algorithm that prioritizes the most effective technique for detecting and classifying spam emails served as the basis for this proposed solution.

4.2 The Proposed Model Architecture

Effective spam detection and classification are essential for maintaining the integrity and usability of email systems. Traditional rule-based filters, while somewhat effective, struggle to keep up with the evolving tactics of spammers. ML models offer a promising solution by automatically learning from data to detect and classify spam more accurately and efficiently. The design suggested a method to enhance detecting and classifying email spam attacks with a better accuracy combines ML with TF-IDF features selection mechanism.

In our proposed model, we choose the most pertinent feature from the datasets needed for our ML model by using a suitable feature selection method, which helps to minimize the complexity as shown in *Figure 4-1*. In addition, it will help to enhance the detection and classification of spam emails by increasing the accuracy and decrease classification errors and training time. Furthermore, to mitigate the possibility of a biased and poorly fitted model, we employ class imbalancing handling strategies like SMOTE in this thesis. In order to successfully detect and categorize spam emails and normal ones, this will improve the model's performance and develop a representative model.

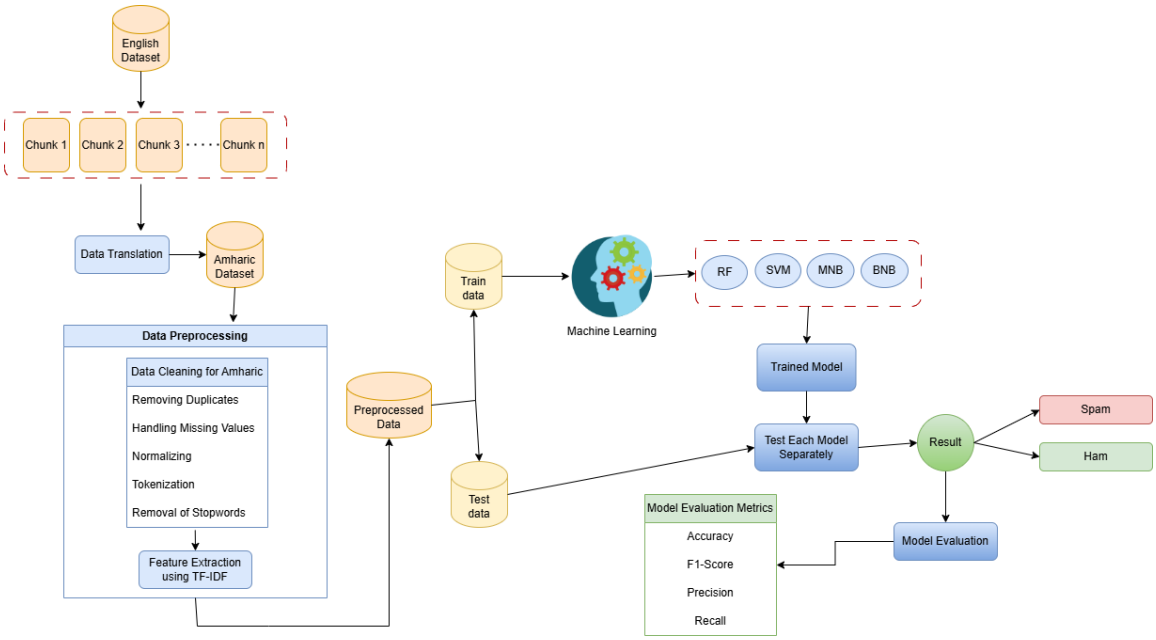


Figure 4- 1: The Proposed Architecture

4.3 Algorithm Flows for Proposed Model

It is necessary to explain the models sequential operations from input to output metrics to more clearly illustrate the structure of the algorithm flow for a proposed ML model. Identifying emails as spam or ham is our task; hence, we employed four distinct training models. An all-purpose common template for an ML models chosen algorithmic flow shown below.

Begin()

Import Libraries:

- Import necessary libraries such as pandas, sklearn, imblearn, and matplotlib.

Load Datasets:

- Load the datasets amharic_df and amharic_df2 using pandas.read_csv().

Merge Datasets:

- Concatenate the datasets into a single DataFrame combined_df.

Preprocess Data:

- Fill missing values in the ‘subject’ column with empty strings.

TF-IDF Vectorization:

- Initialize TfidfVectorizer and fit-transform the ‘subject’ column to create X.
- Extract the ‘label’ column as y.

Balance Dataset:

- Apply SMOTE to balance the dataset, resulting in X_resampled and y_resampled.

K-Fold CV:

- Initialize SVM model.
- Define k-fold CV configuration.
- Perform CV and print scores.

Train-Test Split:

- Split the resampled data into training and testing sets of 0.3.

Train Model:

- Train the SVM model using the training data.

Make Predictions:

- Predict labels for the test data.

Evaluate Model:

- Calculate precision, recall, and F1-score.

- Generate and display the confusion matrix.

Return detection_classification {}
End()

4.4 Optimization of Hyper Parameters

In the realm of ML, the process of optimizing hyper-parameters is crucial for achieving high model performance, particularly in complex tasks such as email spam detection and classification. Hyper-parameters, unlike model parameters, not learned during training but set before the learning process begins. They play a vital role in defining the structure and behavior of a ML model. In the context of our thesis, which focuses on enhancing email spam detection using ML, optimizing hyper-parameters was a critical step in achieving superior model accuracy, precision, recall, and F1-score. We have used cross validation often employed to ensure that the model's performance evaluated accurately. By dividing the data into training and validation sets multiple times, CV helps in selecting hyper-parameters that generalize well to unseen data.

4.4.1 Results of Optimization

The results of hyper-parameter optimization in this research clearly demonstrated the importance of this process in ML. The optimized SVM model not only achieved the highest accuracy but also showed superior performance across other metrics such as precision, recall, and F1-score. This indicates that careful tuning of hyper-parameters can lead to significant improvements in model performance, especially in complex tasks like email spam detection. In addition to achieving high accuracy, the optimization process also helped in reducing overfitting, ensuring that the models generalized well to new, unseen data. This is particularly important in real-world applications, where models must perform reliably on diverse datasets.

Chapter Five

5. Implementation of Proposed Model Architecture

5.1 Introduction

In this section, we looked at how the proposed model used for ML to detect and classify email spam attacks. We also looked at setting up and using the Jupyter Notebook implementation environment.

5.2 Working Environment Setup

Setting up the environment is a crucial step for any researcher setting up the computer environment for tasks involving pattern detection and classification. This includes the different package libraries, tools, and ML structures that used for different purposes, as listed below.

- **Jupyter Notebook**: is an open-source, interactive web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It is widely used for data analysis, ML, and scientific research.
- **Python**: a versatile and powerful programming language that significantly aids research across various fields.
- **Pandas**: a powerful and widely used Python library designed for data manipulation and analysis.
- **Scikit-Learn**: a widely used Python library for ML, offering simple and efficient tools for data mining and data analysis.
- **Matplotlib**: a popular Python library for creating static, animated, and interactive visualizations.
- **re**: allows us to perform complex pattern matching and text manipulation tasks.
- **nltk.corpus.stopwords**: this module from the Natural Language Toolkit (NLTK) provides a list of common stopwords (like "ነው," "ከዚህ," "ላይ," etc.) in English languages.
- **nltk.tokenize.word_tokenize**: this function from NLTK used to split text into individual words, known as tokens.
- **nltk.stem.PorterStemmer**: PorterStemmer is a popular stemming algorithm available in NLTK.

- **nltk.stem.WordNetLemmatizer**: The WordNetLemmatizer uses the WordNet database to return the base form of words, considering the context and the word's meaning, leading to accurate results compared to stemming.

5.3 Implementation of Proposed Solution

5.3.1 Preprocessing

5.3.1.1 Data Collection

The initial step in this research involved the collection of datasets containing both Amharic and English emails. However, Amharic data is not available in public dataset. Therefore, we created our own dataset using the Google translator API. These datasets sourced from public dataset, which is Kaggle.com and local dataset from Ethio Telecom Company, ensuring a diverse range of email types, including spam and legitimate emails. As shown in *Table 3-1* Spam emails labeled as "1" and Legitimate (ham) emails labeled as "0". In addition, we collected Amharic stopwords data from Github repository for preprocessing purpose.

5.3.1.2 Data Translation

Given the bilingual nature of the public and local dataset, it was necessary to translate English and other language emails into Amharic to create a consistent linguistic base for model training. As shown in *Appendix 3*, to create a Python program that reads a CSV file containing English and other language text, translates into Amharic, and then saves the translated content back to a new CSV file, we used the *pandas* library for handling the CSV files and a translation library like *googletrans* and *langdetect* for detecting language and translating the text. The translated data looks as shown in *Table 3-2*.

5.3.1.3 Data Cleaning

Before the data used for training ML models, it was essential to clean the datasets. The data cleaning performed separately for each Amharic corpus due to the 'googletrans' library limitation. The Google translator API can only translate 5000 characters. Amharic uses the Ge'ez script, which has over 300 characters, including syllabic characters. Handling these requires special consideration, especially for tasks like removing punctuation, normalizing text, removing stopwords, and tokenization.

The dataset loaded from a CSV file named such as, “ling_classification.csv” and then removed duplicate rows with rows having missing email contents. In normalization step, all numbers, extra whitespaces, punctuation, and special characters removed. Common stopwords such as “ከዚህ”, “ነው” removed using NLTK stopwords list. Stemming and Lematization used to reduce words to their root form such as “running” to “run”, and converts words to their base form such as, “better” to “good”. But in our case Stemming and Lematization is not supported Amharic language so we skipped this step for good. Finally, the cleaned data saved to a new CSV file named “ling_cleaned_emails.csv”. The cleaned data looks in *Table 5-1*.

The other dataset is also loaded from a CSV file named “emailSubject_amharic.csv” and then duplicate rows and rows with missing email content removed. In normalization step all numbers, extra whitespaces, and non-Amharic punctuation removed. In tokenization and stopwords removal step we tokenize the text and removed stopwords using the list of Amharic stopwords data obtained from Github.com. Finally, the cleaned data saved to a new CSV file named “cleaned_amharic_emails.csv”. The cleaned data looks as shown in *Table 5-1*.

Table 5- 1: Cleaned Amharic Dataset Sample

email	**label**
ለፕሪሚየም ይዘት ባህርያት መዳረሻ ፕሪምየም እቅዳችን አሻሽል	0
መልካም በዓል ከቡድናችን! በዚህ ወቅት ድስታን ብልጽግናን እመኛለሁ	0
እየቀጠርን የስራ እድሎቻችንን ይመልከቱ ተለዋዋጭ ቡድናችንን ይቀላቀሉ	0
የአማዞን መለያህ ተቆልፏል የመለያዎን መረጃ ለማረጋገጥ ጠቅ ያድርጉ	1
የእርስዎ አስተያየት አስፈላጊ የዳሰሳ ጥናታችንን ይውሰዱ ልምድዎን እንድናሻሽል ያግዙን	0
ክፍያዎ ደርሷል ለፈጣን ግብይትዎ እናመሰግናለን	0
የኢሜይል መለያህ ማከማቻ ሙሉ መለያህን ለማሻሻል ጠቅ አድርግ	1
ውድ [ስም]፣ ለጋዜጣችን ደንበኝነት ስለተመዘገቡ እናመሰግናለን የእንኳን ደህና መጣችሁ ስጦታ ይኸውና	0
መለያዎ በታማኝነት ነጥቦች ተቆጥሯል ለአስደናቂ ሽልማቶች	0
ለነጻ iPhone ተመርጠዋል ሽልማትዎን ለማግኘት ጠቅ ያድርጉ	1
የእኛን ልዩ ቅናሽ እንዳያመልጥዎ አሁን ይመዝገቡ በመጀመሪያ ግዢዎ ቅናሽ ያግኙ	0
ለበጋው ተለማማጆችን እየቀጠርን አሁን ያመልክቱ ጠቃሚ ተሞክሮ ያግኙ	0
ለብድር አስቀድመው ተፈቅደዋል ለማመልከት ጠቅ ያድርጉ	1
አዲሱን ስብሰባችንን ስናስተዋውቅ በጣም ደስ ብሎናል አሁን ይግዙ ቅናሾችን ይደሰቱ	0
መጭውን የዌቢናር ተከታታዮቻችንን ለማሳወቅ ጓጉተናል ቦታዎን ለማስያዝ አሁን ይመዝገቡ	0
በአስተያየትዎ መሰረት አዳዲስ ባህርያትን መተግበሪያችን አክለናል አሁን አዘምን	0
እርስዎ አሸናፊ ነዎት ሽልማትዎን ለማግኘት ጠቅ ያድርጉ	1
የፌስቡክ አካውንትህ ተጠልፏል መለያህን ለመጠበቅ ጠቅ አድርግ	1
አዲስ ምዕራፍ ስለደረስዎ እንኳን ደስ አለዎት ለብዙ ተጨማሪ ስኬቶች እነሆ	0

5.3.1.4 Balancing the Datasets

An imbalanced dataset can lead to a model that biased towards the majority class, resulting in poor performance on the minority class. A balanced dataset helps the model generalize better to new unseen data, ensuring it performs well across different classes. We have used sereral DataFrames, such as amharic_df1, amharic_df2, and amharic_df3. Then we merges them into a single DataFrame called combined_df. We have used SMOTE technique to balance the datasets. As shown in *Figure 5-1*, the technique generates synthetic samples for the minority class to ensure an equal distribution of classes.

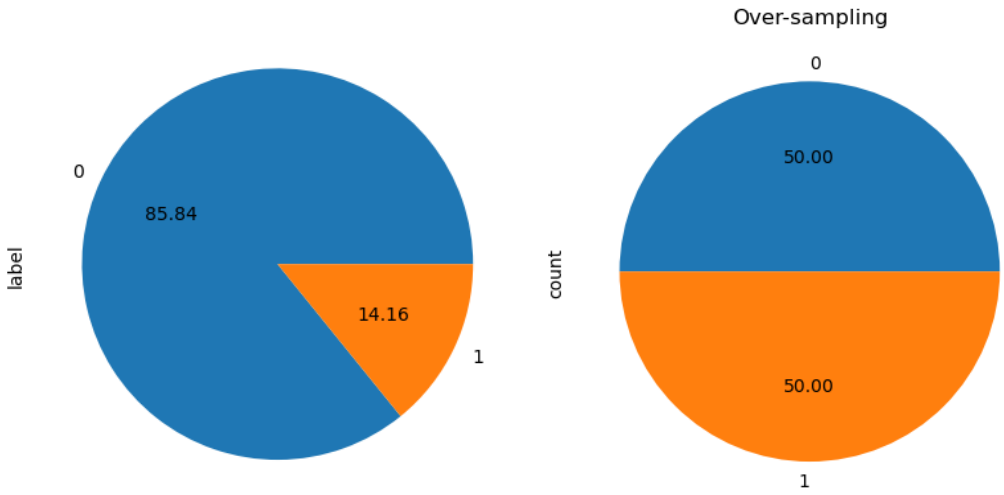


Figure 5- 1: Balance dataset before and after

5.3.1.5 Feature Extraction

In this step, after loading all Amharic cleaned dataset we merged them into a single Data Frame called "combined_df". The TF-IDF Vectorization is used to convert the emails text into a matrix of TF-IDF feature. This matrix converted into a Data Frame for better visualization.

5.3.2 Model Implementation

We discussed model implementation in this section, which is an important scenario in any study topic. Along with the optimization parameters we used for model training, it provides an implementation of the chosen model for Improving Email Spam Detection and Classification through ML Approaches.

5.3.2.1 NB Model Implementation

NB is a simple yet powerful probabilistic classifier based on Bayes' Theorem, assuming independence among features. NB calculates the probability of each class such as, spam or not

spam given a set of features such as, word frequencies and assigns the class with the highest probability to the new data. In our research on Enhancing Email Spam Detection and Classification using ML with Amharic datasets, NB particularly effective. It is well suited for text data because it handles high-dimensional inputs efficiently and works well even with relatively small datasets. The algorithm's simplicity allows for quick training and prediction, which is beneficial when dealing with large volumes of emails. Additionally, its performance is often competitive with other models, especially when features are appropriately pre-processed, such as through the removal of stopwords or the transformation of text data into term frequency-inverse document frequency (TF-IDF) vectors. Incorporating NB into our model provide a strong baseline or even a robust standalone solution, particularly when combined with the multilingual nature of your dataset. We trained both the MNB and BNB models using the dataset. These models selected due to their effectiveness in text classification tasks, particularly in spam detection. The training process involved splitting the dataset into training and testing sets, followed by fitting the models to the training data. The dataset utilized for this study contains 7053 email records which is 1554 records from local (Ethio Telecom) and 5499 from public dataset, where translated into Amharic emails. This diverse dataset was crucial in building a model capable of handling multilingual email detection and classification tasks. To assess the performance of both MNB and BNB model, we employed Accuracy, Precision, Recall, and F1-score.

5.3.2.2 SVM Model Implementation

SVM is a powerful supervised learning algorithm commonly used for classification tasks. In our research, SVM helps by creating a hyperplane or decision boundary that best separates the different classes in the dataset such as, spam and non-spam emails. SVM is particularly effective when dealing with high-dimensional data, like text, because it focuses on finding the optimal margin between classes, leading to better generalization and accuracy. By using SVM, we capture complex relationships in Amharic datasets, potentially improving the model's ability correctly classify emails, even in cases where the data points are not linearly separable. This can be especially beneficial in detecting subtle patterns in spam emails that missed by other models. We utilized a dataset of 7053 email records which is 1554 records from local (Ethio

Telecom) and 5499 from public dataset, comprising all Amharic emails, to train the SVM model.

The training process involved splitting the dataset into training and testing sets. The SVM model then trained on the training set, where it learned to identify the optimal decision boundary between spam and non-spam emails. The goal is accurately classify emails as either spam or legitimate using the SVM's robust capabilities in handling high-dimensional data. The performance of the SVM model assessed using Accuracy, Precision, Recall, and F1-score as well.

5.3.2.3 RF Model Implementation

RF is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes for classification tasks. In our research, incorporating RF provide several advantages. This model helps by reducing the risk of overfitting, which is common in individual decision trees, especially when dealing with complex datasets like those in Amharic and English. RF improves classification accuracy by averaging the results of multiple trees, each trained on random subsets of the data and features, which leads to better generalization to unseen data. This ensemble approach is particularly beneficial in handling noisy data and capturing diverse patterns in emails, making it a robust choice for improving email detection and classification performance. We trained the RF model on the dataset to classify emails into spam and non-spam categories. The RF algorithm chosen due to its ability to handle a large number of features and its robustness against overfitting. It operates by constructing a multitude of decision trees during training and outputting the mode of the classes (for classification) of the individual trees. The dataset used for this study comprises 7053 email records which is 1554 records from local (Ethio Telecom) and 5499 from public dataset, featuring content in Amharic. This diverse dataset allows the model to learn and adapt to the nuances of multiple languages, thereby enhancing its ability accurately classify emails. The training process involved splitting the dataset into training and testing sets. The RF model then trained on the training set, where it learned to identify patterns and relationships within the data that differentiate spam emails from non-spam emails. The performance of the RF model evaluated using Accuracy, Precision, Recall, and F1-score.

5.4 Model Testing and Evaluation

After model implementation, its performance assessed using the resampling approach known as CV. CV is a crucial technique in ML used to evaluate the performance of a model and ensure its generalizability to unseen data. This research reviewed various CV methods, their advantages, and their application in model evaluation. A popular method in machine learning for evaluating a model's performance and generalizability is 5-fold CV. The dataset divided using this method into five equal subsets, or “folds”. In order to ensure that each fold used as the validation set once, the model trained on four folds and validated on the fifth.

CV helps mitigate overfitting and provides a more reliable estimate of model performance compared to a simple train-test split. K-Fold CV is a widely used method where the dataset divided into K equally sized folds. The model is trained K times, for training and the remaining fold for testing. This process repeated K times, with each fold serving as the test set exactly once. The final performance metric averaged over all K iterations.

Chapter Six

6. Result and Discussion

The outcomes of the suggested methodology for improving email spam detection and classification covered in this chapter. Our suggested models are MNB, BNB, SVM, and RF, thus, their performance evaluated in order to determine which model performs best in terms of spam email detection and classification.

6.1 Result of the Proposed Models

Our experiment, which built for ML models covered here. For improving email spam detection and classification employing spam email corpus for each of the models, MNB, BNB, SVM, and RF were the test models. Every model that tested, evaluated, and the results displayed below, section by section, utilizing statistically chosen criteria to identify spam activity. Each model utilized 7053 total records Amharic language data. Finally, based on evaluation measures, we selected a model that provides the best performance for the given tasks.

6.1.1 MNB Model Experiment

The goal of the experiment to evaluate the effectiveness of the MNB model in classifying emails as spam or ham (not-spam) involved using all Amharic datasets, which added a multilingual dimension to the analysis. The model trained on the cleaned and preprocessed datasets.

In TF-IDF vectorization *max-features* parameter limits the number of features (words) extracted from the text data setting to 5000. In *train_test_split* we used fixed *radom_state* setting to 42 and *SMOTE* used to control oversampling. In incremental learning *batch_size* parameter set to 100 to control the size of data batches used for partial fitting. The experiment employed K-fold cross validation. A 5-fold cross validation performed to assess the model's performance on different subsets of the data. The CV scores for each fold were as shown in *Table 6-1*. The average accuracy across the folds was **0.9268** with a standard deviation of **0.0042**, indicating consistent performance across the different splits. The consistent CV scores suggest that the MNB model generalizes well to unseen data. The low standard deviation further supports the model's reliability. The dataset first resampled to address any potential class imbalance. The data then split into 70% for training and 30 % for testing sets using the *'train_test_split'*

function. MNB model was trained using the *'MultinomialNB()'* class from the *'sklearn'* library. The training process involved fitting the model to the training data *'X_train', 'y_train'*.

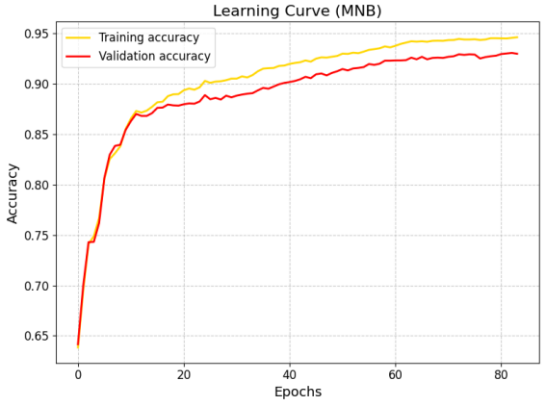


Figure 6- 1 Training and Validation Accuracy of MNB

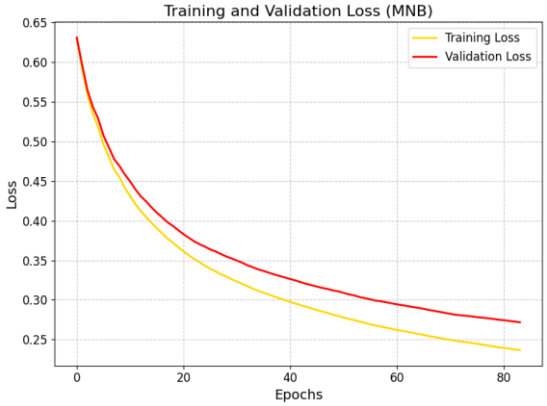


Figure 6- 2 Training and Validation Loss of MNB

6.1.2 BNB Model Experiment

The BNB model chosen for this experiment due to its suitability for binary feature sets, which are common in text classification tasks like spam detection. Since BNB is inherently designed to handle binary data, it naturally fits the scenario where the features are represented such as, 0s and 1s the binary presence or absence of words in an email. This alignment between the model's expectations and the data's structure leads to more effective learning and classification. The dataset was carefully prepared, involving resampling techniques using *SMOTE* to balance the classes, followed *train_test_split* by splitting the data into training and testing sets. This ensures that the model trained on a representative sample and evaluated on unseen data. CV is a critical step in assessing the generalization ability of a model. The BNB model underwent 5-fold CV, yielding the following scores for each fold shown in *Table 6-1*. These scores are consistently high, with an average accuracy of **0.9034** and a standard deviation of **0.0054**. The standard deviation indicates that the model performs uniformly across different data splits, highlighting its robustness and reliability. The BNB experiment has demonstrated the model's good performance in spam detection tasks.

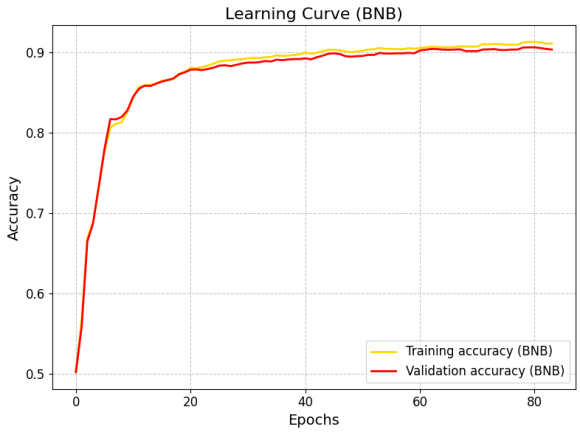


Figure 6- 3 Training and Validation Accuracy of BNB

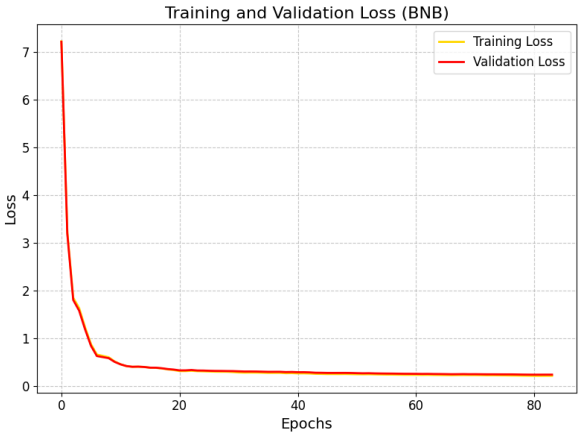


Figure 6- 4 Training and Validation Loss of BNB

6.1.3 SVM Model Experiment

The SVM model selected for the experiment to evaluate its effectiveness in classifying emails as spam or ham. SVM is known for its strong performance in binary classification tasks, particularly when the data is high dimensional, as is often the case in text classification. The dataset included emails written in Amharic.

The text data preprocessed, including tokenization, vectorization, and potentially applying techniques such as, TF-IDF to convert the text into numerical features suitable for the SVM model. The dataset then split into training and testing sets, with 70% used for training the model and 30% reserved for evaluating its performance.

To assess the model's generalization ability, a 5-fold CV performed, yielding the results as shown in *Table 6-1* where the Average Accuracy **0.9852** and Standard deviation **0.0028** indicates that SVM model is both effective and stable, in its performance across different subsets of the data. The SVM experiment confirms the model's strength in email classification tasks; achieving high accuracy and balanced performance across both spam and ham classes.

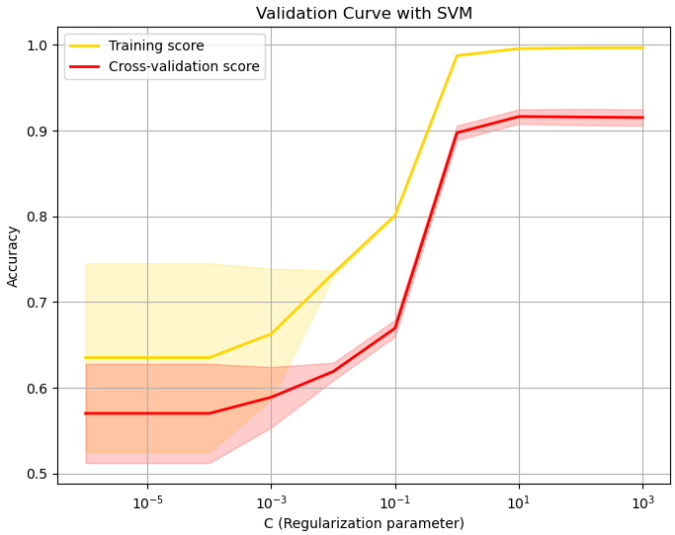


Figure 6- 5 Validation Curve of SVM

The examination of an SVM model's validation curve in *Figure 6-5* reveals how the model's performance changes as the regularization parameter C is varied. The training score represents the model's accuracy on the training data. As C increases, the model tends to fit the training data more closely, potentially leading to higher training scores. This is because a higher C value reduces the regularization strength, allowing the model to capture more complex patterns in the training data. The cross validation score indicates the model's performance on unseen data, assessed through CV. The CV score helps in understanding how well the model generalizes to new data. The validation curve analysis helps in identifying the appropriate regularization strength for the SVM model to achieve a balance between underfitting and overfitting, ultimately leading to better generalization performance.

6.1.4 RF Model Experiment

The RF model employed to evaluate its performance in classifying emails as either spam or ham. RF is a versatile and widely used ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) of the individual trees. The method is especially valued for its ability to handle large datasets and its resistance to overfitting due to its use of multiple decision trees.

The experiment conducted using a dataset of 7053 records which is 1554 records from local (Ethio Telecom) and 5499 from public dataset, comprising emails written in Amharic. The data

underwent preprocessing, including TF-IDF vectorization to transform the textual data into a suitable numerical format for the RF model.

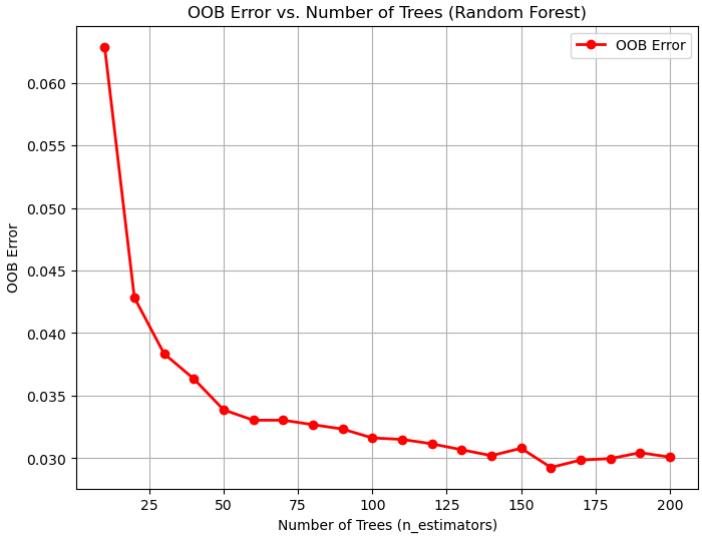


Figure 6- 6 OOB Error result of RF

The dataset then divided into training (70%) and testing (30%) sets, with a *'random_state'* set to 42 to ensure reproducibility. To evaluate the generalization capability of the RF model, a 5-fold CV performed, resulting as shown in *Table 6-1* where Average Accuracy **0.9655** and Standard Deviation **0.0047** indicates that the model performs well consistently across different folds. For RF, as shown in *Figure 6-6* we track the Out-of-Bag (OOB) error during training as a proxy for model performance. It shows how the error decreases as more trees added to the forest.

Table 6- 1: K-fold cross validation results using 5-folds

Metrics	**5-Folds Cross Validation**				
	1	**2**	**3**	**4**	**5**
MNB	0.932	0.926	0.931	0.92	0.925
BNB	0.904	0.909	0.908	0.896	0.897
SVM	**0.983**	**0.981**	**0.988**	**0.984**	**0.988**
RF	0.967	0.969	0.967	0.966	0.956

6.2 Results of Proposed Models

In this research, we evaluate the performance of four different ML models where MNB, BNB, SVM, and RF for the task of classifying emails into spam and ham (non-spam). The dataset

used contains 7053 records which is 1554 records from local (Ethio Telecom) and 5499 from public dataset, in Amharic, and the goal is to identify the strengths and weaknesses of each model in handling diverse dataset. The dataset balanced using SMOTE technique to use equal class for model training. In addition, we have used 70% for training and 30% for testing using 'random_state = 42' in all models for comparison. All model measured using K-fold cross validation technique utilizing 5-folds as shown in *Table 6-1*.

6.2.1 Proposed Models Evaluation Results

In the evaluation of ML models, several classifiers assessed based on their accuracy, precision, recall, and F1-score. Among the models tested, SVM demonstrated the highest performance with an accuracy of 98.52%. It also excelled in precision, recall, and F1-score, achieving 97.88% each. This indicates that SVM not only correctly classified the majority of the emails but also maintained a balanced performance across both spam and non-spam categories.

The BNB model followed with an accuracy of 90.34%. BNB also showed strong results in precision, recall, and f1-score with values of 91.45%, 90.56% and 90.51%, respectively. The MNB model had an accuracy of 92.68%, which is a little bit lower than both BNB and SVM. It achieved a precision of 93.01% and a recall of 93.01%, with an F1-score of 93.01%. While MNB's performance was commendable, it did not match the higher levels of accuracy and balanced performance provided by SVM and BNB.

The RF model exhibited the lowest performance among the classifiers, with an accuracy of 96.55%. Precision was 96.68%, recall was 96.67%, and the F1-score was 96.67%. Although RF had good precision, its lower recall and F1-score compared to the other models suggest it struggled more with maintaining performance across both classes.

Table 6- 2: The Proposed Models Evaluation Metrics of the Performance

Metrics		Accuracy	Precision	Recall	F1-socre
Model	MNB	92.68%	93.01%	93.01%	93.01%
	BNB	90.34%	91.45%	90.56%	90.51%
	SVM	**98.52%**	**97.88%**	**97.88%**	**97.88%**
	RF	96.55%	96.68%	96.67%	96.67%

As shown in *Table 6-2*, the SVM model outperformed the others in accuracy and balanced classification performance, making it the most effective model for email spam detection in this evaluation. BNB followed with strong results, while MNB and RF showed lower performance in comparison.

6.2.2 Confusion Matrix Classification Results

In *Figure 6-7*, the confusion matrix for MNB, BNB, RF, and SVM experimented with classification results for a set of instances. Each confusion matrix provides insights into the accuracy of the models by indicating the number of true positives (correctly identified), true negatives, false positives (incorrectly identified), and false negatives from 7053 total data. MNB model correctly classified 1688 instances as spam emails and 1691 as legitimate emails, while 129 emails misclassified as legitimate emails and 125 emails as spam emails. This indicates the MNB model classified well to detect and classify spam emails from legitimate emails.

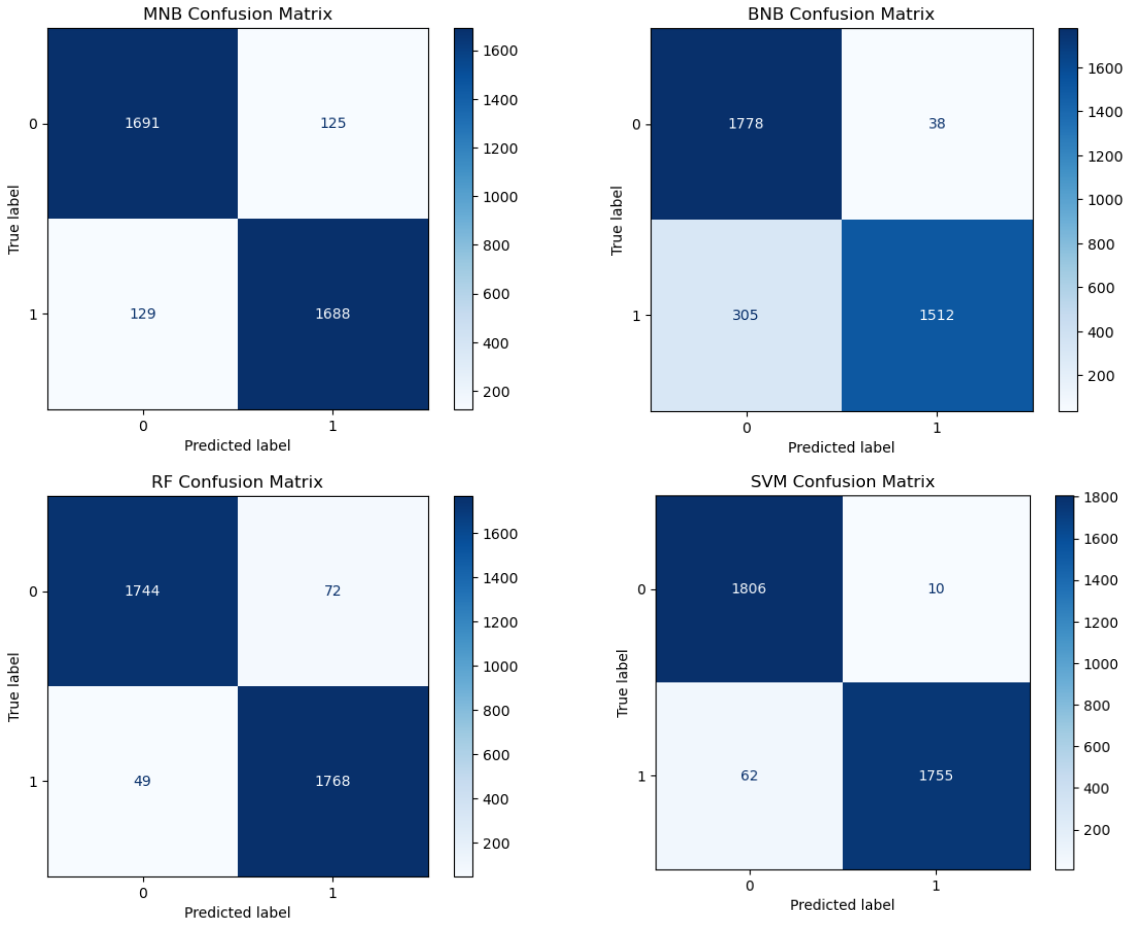


Figure 6- 7 Confusion Matrix results using MNB, RF, SVM and BNB models respectively

Similarly, BNB model measured using confusion matrix as shown in *Figure 6-7*. From 7053 total data, BNB model correctly classified 1512 emails as spam emails and 1778 as legitimate emails, while 305 emails misclassified as legitimate and 38 emails as spam. This indicates the BNB model classified better than MNB by minimizing misclassified data and by increasing correctly classified data as well.

Similarly, for RF model the confusion matrix measured for a set of instances as shown in *Figure 6-7*. The RF model classified correctly 1768 emails as spam and 1744 as ham (legitimate) emails. However, it misclassified 49 emails as legitimate emails and 72 as spam emails.

The proposed SVM model similarly measured using confusion matrix for a set of instances as shown in Figure 6-1 classifying correctly 1755 emails as spam and 1806 as legitimate emails, while 62 emails misclassified as legitimate and 10 emails as spam. The SVM performs very

well with the lowest number of false positive, indicating high precision. However, it has a slightly higher number of false negative compared to MNB, meaning it occasionally misses some positive cases.

6.3 Comparison of Evaluation Metrics

Here, we go over the evaluation metric's experimental findings utilizing 5-fold Cross Validation and MNB, BNB, SVM, and RF. Based on their evaluation performance, this assisted in choosing the optimal model to improve email spam detection and categorization in an efficient manner. Consequently, MNB, BNB, SVM, and RF are the three models trained on the email spam detection and classification environments, as shown in *Figure 6-8*. As a result, SVM outperforms MNB, BNB, and RF in *Figure 6-8* for all evaluation metrics.

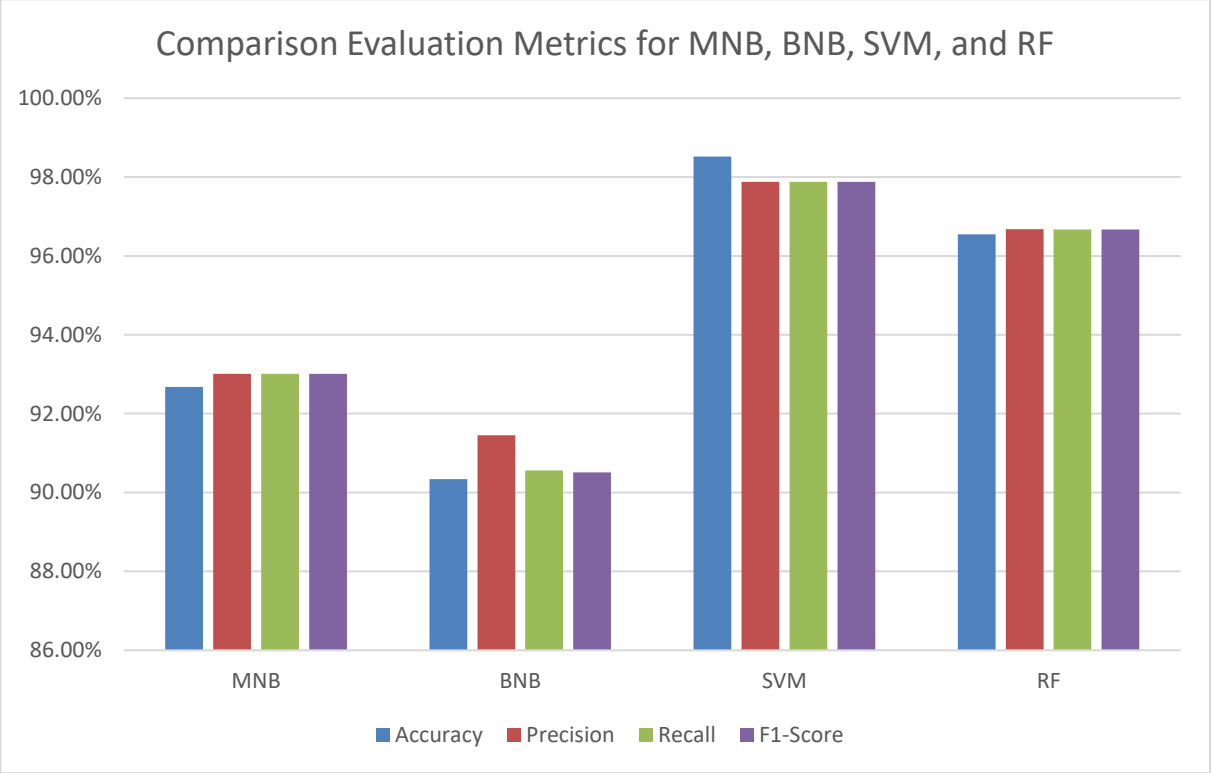


Figure 6- 8: Comparison Evaluation Metrics

6.4 Discussion of Research Question

Four research questions, located in the first chapter of section 1.4, addressed in this study. As a result, our research has provided the following sequence of answers to this question.

RQ1: How the existing non-English dataset used for Email spam detection and classification?

We studied the existing email spam detection and classification methods identifying their strengths and weaknesses, exploring potential areas for improvement. We start our study by identifying recent research where non-English dataset in the field of email spam detection. From the existing data we analyze (Siddique et al., 2021) article that train ML model using Urdu language text data where translated from English using Google translator API. When ML algorithms utilized such as NB, CNN, SVM, and LSTM to detect and classify spam emails written in Urdu language. The LSTM model achieved the highest accuracy outperforming other models in detecting Urdu spam emails. Similarly, (Rastenis et al., 2021) developed a solution for classifying emails written in English, Russian, and Lithuanian by translating existing English datasets into these languages using automated tools like Google Translate. Their approach demonstrated that such translations could effectively augment datasets for multilingual spam detection.

RQ2: How a new Amharic dataset can be created using Google translator API?

To create a new Amharic dataset using the Google Translate API, an existing English email dataset, such as Kaggle and Ethio Telecom data used as a source. First we collected the dataset in a structured format, typically in CSV, containing email content and corresponding labels (spam or ham). Using the *googletrans* library in Python, each email text then translated into Amharic by sending API requests to Google Translate. Since the API has character limitations per request, large text data splited into smaller chunks before translation. Once translated, the Amharic dataset reconstructed and stored while preserving the original labels. After translation, standard preprocessing steps like stopword removal, punctuation handling, and text normalization applied to refine the dataset. The newly created Amharic dataset then used for ML experiments in email spam detection, enabling the development of models specifically tailored to the Amharic language.

RQ3: How ML model can be developed to enhance the performance of Email spam detection and classification?

Developing ML model to enhance the performance of email spam detection and classification requires a well-structured pipeline that includes data preparation, feature extraction, model selection, optimization, and evaluation. The process begins with collecting a high-quality dataset, such as Ethio Telecom and Kaggle, then creating a new dataset in a specific language like Amharic using Google Translate API. Preprocessing techniques, including tokenization, stopword removal, lemmatization, and handling imbalanced data with techniques like SMOTE, are essential for improving model performance. Feature extraction methods such as Term Frequency-Inverse Document Frequency (TF-IDF) help convert text data into numerical representations suitable for ML models. Choosing the right classification model is crucial. Traditional algorithms like NB, SVM, and RF models used for enhanced accuracy. Finally, evaluating the model using metrics like accuracy, precision, recall, and F1-score ensures a comprehensive understanding of its effectiveness.

RQ4: How to evaluate the performance of the proposed solution?

Evaluating the performance of ML models is crucial to understand how well the model is performing and to compare it with other models or baselines. The evaluation process typically involves using a variety of metrics and techniques to assess the model's accuracy, robustness, and generalizability. In train-test split step, the dataset divided into two parts where a training and testing set. The model trained on the training set and evaluated on the test set. In CV step, the dataset divided into ‘k’ subsets (folds). The proposed model trained on 5 folds and tested on the remaining fold. This process repeated ‘k’ times with each fold used as the test set once. Then average result printed. This provides a more reliable estimate of model performance by reducing the variability that might result from a single train-test split. The evaluation metrics used in this research are accuracy, precision, recall, and f1-score including confusion matrix.

Chapter Seven

7. Conclusion and Future Works

7.1 Conclusion

Detecting and classifying email spam is an essential part of cybersecurity because cyber attacks that use email as a key vector becoming more sophisticated. Spam emails are more than just an inconvenience; they frequently contain malware, phishing attempts, and other threats that can jeopardize an organization's or individual's security. Data breaches, financial loss, and other cyber disasters are less likely when these threats filtered out before they reach the end user by means of efficient spam detection and categorization systems. Thus, there is a connection between spam detection and cybersecurity since strong spam filters are a first line of defense against many types of cybercrime. Exploiting flaws in language processing is one of the ways spammers keep improving their strategies to get past current spam filters. Being the most frequently used language in spam, English optimized for by the majority of spam detection systems. To avoid detection, spammers are increasingly using non-English languages, including those that are less widely spoken, in their emails. Due to the detection systems' limited linguistic capabilities, these emails are more likely to evade the filters even though they might contain the same harmful information. The necessity for more sophisticated and multilingual spam detection algorithms that can identify and filter spam in a wider variety of languages is highlighted by this difficulty, which will help to bridge the gap that spammers take advantage of. This study successfully illustrates how using the Google Translate API to translate English datasets improves email spam detection and classification for Amharic data. The data was carefully cleaned, and stopwords were eliminated using a list of Amharic stopwords that was downloaded from GitHub. Four ML models MNB, BNB, SVM, and RF trained and assessed using TF-IDF techniques for feature extraction and the SMOTE technique to address dataset imbalance. According to the findings, the MNB model's accuracy was 92.68%, BNB's 90.34%, SVM's 98.52%, and RF's 96.55%. When it came to identifying and categorizing Amharic emails for spam detection, the SVM model performed better than the others. In the case of Amharic, this study emphasizes the SVM model's potential as a powerful tool for improving email security in languages such as Amharic.

7.2 Contribution of the Study

- For communities that speak Amharic, improving spam filtering for the language can make digital communication safer and easier.
- As a reference for next studies, compile a large dataset of Amharic emails that classified as spam or not.
- Examining Amharic spam patterns can assist identify regional cyber threats and guide the creation of focused cybersecurity plans.
- Increase the effectiveness of feature engineering techniques by identifying distinctive patterns and characteristics in Amharic emails that are suggestive of spam.
- Better spam filters reduce the incidence of phishing and other malicious activities in Amharic, contributing to overall cybersecurity.

7.3 Future Works

NLP and ML advancements have produced new frameworks and tools for more reliable and effective spam identification. Comprehending and incorporating these contemporary technologies can greatly improve the system's performance. Several approaches should be investigated in future research to improve the efficacy of email spam detection and categorization, especially for non-English languages like Amharic. First, better model performance and richer linguistic features could be obtained by creating a more complete and native Amharic dataset instead of depending on translated data. Incorporating deep learning methods, like transformers or neural networks, may also increase accuracy by identifying intricate patterns in the data. Developing a more comprehensive list of Amharic stopwords and examining the effects of alternative feature extraction techniques outside of TF-IDF two more possible avenues for improving the preprocessing stages.

Furthermore, experimenting with ensemble learning approaches or incorporating and testing more sophisticated oversampling strategies could improve the model's capacity to handle unbalanced datasets. Finally, broadening the scope of the research to incorporate additional languages and a wider range of datasets may aid in the creation of generalizable models that can better identify and categorize spam in various linguistic situations. In today's globalized world, where spammers are using non-English languages more frequently to avoid detection, this would be especially helpful.

Special Acknowledgement

This research work was funded by Adama Science and Technology University under grant number

ASTU/SM-R/1062/24

Adama, Ethiopia

References

Afzal, H., & Mehmood, K. (2016). Spam filtering of bi-lingual tweets using machine learning. *2016 18th International Conference on Advanced Communication Technology (ICACT)*, 710–714. https://doi.org/10.1109/ICACT.2016.7423530

Ahmed, M. T., Akter, M., Rahman, M. S., Rahman, M., Paul, P. C., Parvin, M. N., & Antar, A. H. (2023). Deep Neural Network Based Spam Email Classification Using Attention Mechanisms. *Journal of Intelligent Learning Systems and Applications*, *15*(04), 144–164. https://doi.org/10.4236/jilsa.2023.154010

Alavi, R., Islam, S., & Mouratidis, H. (2016). An information security risk-driven investment model for analysing human factors. *Information & Computer Security*, *24*(2), 205–227. https://doi.org/10.1108/ICS-01-2016-0006

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the study of SMS spam filtering: new collection and results. *Proceedings of the 11th ACM Symposium on Document Engineering*, 259–262. https://doi.org/10.1145/2034691.2034742

Altulaihan, E., Alismail, A., Hafizur Rahman, M. M., & Ibrahim, A. A. (2023). Email Security Issues, Tools, and Techniques Used in Investigation. *Sustainability (Switzerland)*, *15*(13). https://doi.org/10.3390/su151310612

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V, & Spyropoulos, C. D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 160–167. https://doi.org/10.1145/345508.345569

Arya, V., Almomani, A. A. D., Mishra, A., Peraković, D., & Rafsanjani, M. K. (2023). *Email Spam Detection Using Naive Bayes and Random Forest Classifiers BT - International Conference on Cyber Security, Privacy and Networking (ICSPN 2022)* (N. Nedjah, G. Martínez Pérez, & B. B. Gupta (Eds.); pp. 341–348). Springer International Publishing.

Assegie, T. A. (2023). *Evaluation of Supervised Learning Models for Automatic Spam Email Detection*. 1–10.

Balduzzi, M., Platzer, C., Holz, T., Kirda, E., Balzarotti, D., & Kruegel, C. (2010). Abusing

Social Networks for Automated User Profiling. In S. Jha, R. Sommer, & C. Kreibich (Eds.), *Recent Advances in Intrusion Detection* (pp. 422–441). Springer Berlin Heidelberg.

Bobde, S., Role, S., Khadke, L., Shirude, T., & Kakad, S. (2023). Email Spam Detection Using Hybridization of SVM and Random Forest. *International Journal of Research in Engineering and Science (IJRES) ISSN*, *11*(7), 188–193. www.ijres.org

Cahyani, D., & Patasik, I. (2021). Performance comparison of TF-IDF and Word2Vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, *10*, 2780–2788. https://doi.org/10.11591/eei.v10i5.3157

Collins, M. (2003). Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, *29*(4), 589–637. https://doi.org/10.1162/089120103322753356

Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, *5*(6). https://doi.org/10.1016/j.heliyon.2019.e01802

De Luna, R. G., Magnaye, V. C., Reaño, R. A. L., Enriquez, K. L., Astorga, D., Celestial, T., Espanola, A. M., Lanting, B. A., Mugar, D., Ramos, M., & Redondo, J. (2023). A Machine Learning Approach for Efficient Spam Detection in Short Messaging System (SMS). *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 53–58. https://doi.org/10.1109/TENCON58879.2023.10322491

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Preprint ArXiv:1810.04805*.

Diyasa, I. (2023). Implementation Of Natural Language Processing for Spam Email Detection in Outcome Based Education (OBE) Application. *IJEBD (International Journal of Entrepreneurship and Business Development)*, *6*, 1166–1171. https://doi.org/10.29138/ijebd.v6i6.2587

Douzi, S., AlShahwan, F. A., Lemoudden, M., & Ouahidi, B. El. (2020). Hybrid Email Spam Detection Model Using Artificial Intelligence. *International Journal of Machine Learning and Computing*, *10*(2), 316–322. https://doi.org/10.18178/ijmlc.2020.10.2.937

Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2008). A survey on automated dynamic

malware-analysis techniques and tools. *ACM Comput. Surv.*, *44*(2). https://doi.org/10.1145/2089125.2089126

Ezenwobodo, & Samuel, S. (2022). International Journal of Research Publication and Reviews. *International Journal of Research Publication and Reviews*, *04*(01), 1806–1812. https://doi.org/10.55248/gengpi.2023.4149

Ferreira, A., Coventry, L., & Lenzini, G. (2015). Principles of Persuasion in Social Engineering and Their Use in Phishing. In T. Tryfonas & I. Askoxylakis (Eds.), *Human Aspects of Information Security, Privacy, and Trust* (pp. 36–47). Springer International Publishing.

Fumera, G., Pillai, I., & Roli, F. (2006). Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research*, *7*(12).

Ghogare, P. P., & Dawoodi, H. H. (2023). *Enhancing Spam Email Classi cation Using Effective Preprocessing Strategies and Optimal Machine Learning Algorithms*. 0–29.

Ghosh, A., & Senthilrajan, A. (2023). *Comparison of machine learning techniques for spam detection Content courtesy of Springer Nature , terms of use apply . Rights reserved . Content courtesy of Springer Nature , terms of use apply . Rights reserved .* 29227–29254.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Holt, T. J., & Graves, D. C. (2007). A qualitative analysis of advance fee fraud e-mail schemes. *International Journal of Cyber Criminology*, *1*(1), 137–154.

Hong, J. (2012). The state of phishing attacks. *Commun. ACM*, *55*(1), 74–81. https://doi.org/10.1145/2063176.2063197

Ismail, S. S. I., Mansour, R. F., Abd El-Aziz, R. M., & Taloba, A. I. (2022). Efficient E-Mail Spam Detection Strategy Using Genetic Decision Tree Processing with NLP Features. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/7710005

Jakobsson, M. (2005). Modeling and preventing phishing attacks. *Financial Cryptography, 5,*

1–19.

Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). A review of spam email detection: analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*, *56*(2), 1145–1173. https://doi.org/10.1007/s10462-022-10195-4

Kaddoura, S., Alfandi, O., & Dahmani, N. (2020). A Spam Email Detection Mechanism for English Language Text Emails Using Deep Learning Approach. *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 193–198. https://doi.org/10.1109/WETICE49692.2020.00045

Kaliappan, J., Bagepalli, A. R., Almal, S., Mishra, R., Hu, Y.-C., & Srinivasan, K. (2023). Impact of Cross-Validation on Machine Learning Models for Early Detection of Intrauterine Fetal Demise. *Diagnostics (Basel, Switzerland)*, *13*(10). https://doi.org/10.3390/diagnostics13101692

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access*, *7*, 168261–168295. https://doi.org/10.1109/ACCESS.2019.2954791

Katsiampa, P. (2019). An empirical investigation of volatility dynamics in the cryptocurrency market. *Research in International Business and Finance*, *50*, 322–335. https://doi.org/https://doi.org/10.1016/j.ribaf.2019.06.004

Keskin, S., & Sevli, O. (2024). Machine Learning Based Classification for Spam Detection. *Sakarya University Journal of Science*, *28*(2), 270–282. https://doi.org/10.16984/saufenbilder.1264476

Madhavan, M. V., Pande, S., Umekar, P., Mahore, T., & Kalyankar, D. (2021). Comparative Analysis of Detection of Email Spam With the Aid of Machine Learning Approaches. *IOP Conference Series: Materials Science and Engineering*, *1022*(1), 12113. https://doi.org/10.1088/1757-899X/1022/1/012113

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? *CEAS*, *17*, 28–69.

Mohamad, M., & Selamat, A. (2015). An evaluation on the efficiency of hybrid feature selection in spam email classification. *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, 227–231.

https://doi.org/10.1109/I4CT.2015.7219571

Mohammad, R. M. A. (2024). A lifelong spam emails classification model. *Applied Computing and Informatics*, *20*(1/2), 35–54. https://doi.org/10.1016/j.aci.2020.01.002

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Olatunji, S. O. (2019). Improved email spam detection model based on support vector machines. *Neural Computing and Applications*, *31*(3), 691–699. https://doi.org/10.1007/s00521-017-3100-y

Ouyang, T., Ray, S., Allman, M., & Rabinovich, M. (2014). A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise. *Computer Networks*, *59*, 101–121. https://doi.org/https://doi.org/10.1016/j.comnet.2013.08.031

Pandey, S., Taralekar, A., Yadav, R., Deshmukh, S., & Suryavanshi, P. S. (2020). E-mail Spam Detection and Classification using SVM. *International Journal of Computer Science and Information Technologies*, *10*(1), 6–8. www.ijcsit.com

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135. https://doi.org/10.1561/1500000011

Paswan, M. K., Shanthi Bala, P., & Aghila, G. (2012). Spam filtering: Comparative analysis of filtering techniques. *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, 170–176.

Professor, A. (2021). Email Spam Detection using Machine Learning and Neural Networks. *International Research Journal of Engineering and Technology*, 349–355. www.irjet.net

Prusty, S., Patnaik, S., & Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, *4*. https://doi.org/10.3389/fnano.2022.972421

Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*. https://api.semanticscholar.org/CorpusID:53702508

Rao, J. M., & Reiley, D. H. (2012). The Economics of Spam. *Journal of Economic Perspectives*, *26*(3), 87–110. https://doi.org/10.1257/jep.26.3.87

Rastenis, J., Ramanauskaitė, S., Suzdalev, I., Tunaitytė, K., Janulevičius, J., & Čenys, A.

(2021). Multi-language spam/phishing classification by email body text: Toward automated security incident investigation. *Electronics (Switzerland)*, *10*(6), 1–10. https://doi.org/10.3390/electronics10060668

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-Validation. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems* (pp. 532–538). Springer US. https://doi.org/10.1007/978-0-387-39940-9_565

Reyns, B. W., Henson, B., & Fisher, B. S. (2011). Being Pursued Online: Applying Cyberlifestyle–Routine Activities Theory to Cyberstalking Victimization. *Criminal Justice and Behavior*, *38*(11), 1149–1169. https://doi.org/10.1177/0093854811421448

Sadia, A., Bashir, F., Khan, R. Q., Bashir, A., & Khalid, A. (2023). Comparison of Machine Learning Algorithms for Spam Detection. *Journal of Advances in Information Technology*, *14*(2), 178–184. https://doi.org/10.12720/jait.14.2.178-184

Sanz, E. P., Gómez Hidalgo, J. M., & Cortizo Pérez, J. C. (2008). Chapter 3 Email Spam Filtering. In *Software Development* (Vol. 74, pp. 45–114). Elsevier. https://doi.org/https://doi.org/10.1016/S0065-2458(08)00603-7

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Shamoo, Y. (2024). *Using Natural Language Processing (NLP) for Phishing and Spam Detection* (pp. 55–78). https://doi.org/10.4018/979-8-3373-0588-2.ch003

Sharaff, A., Nagwani, N. K., & Dhadse, A. (2016). Comparative Study of Classification Algorithms for Spam Email Detection. In N. R. Shetty, N. H. Prasad, & N. Nalini (Eds.), *Emerging Research in Computing, Information, Communication and Applications* (pp. 237–244). Springer India.

Sheneamer, A. (2021). Comparison of Deep Learning and Traditional Methods for Email Spam Filtering. *International Journal of Advanced Computer Science and Applications*, *12*(1), 560–565. https://doi.org/10.14569/IJACSA.2021.0120164

Siddique, Z. Bin, Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine Learning-Based Detection of Spam Emails. *Scientific Programming*, *2021*. https://doi.org/10.1155/2021/6508784

Sonare, B., Dharmale, G. J., Renapure, A., Khandelwal, H., & Narharshettiwar, S. (2023a). E-

mail Spam Detection Using Machine Learning. *2023 4th International Conference for Emerging Technology, INCET 2023*, *10*(1), 2658–2664. https://doi.org/10.1109/INCET57972.2023.10170187

Sonare, B., Dharmale, G. J., Renapure, A., Khandelwal, H., & Narharshettiwar, S. (2023b). E-mail Spam Detection Using Machine Learning. *2023 4th International Conference for Emerging Technology, INCET 2023*, *7*(12). https://doi.org/10.1109/INCET57972.2023.10170187

Sonare, B., Dharmale, G. J., Renapure, A., Khandelwal, H., & Narharshettiwar, S. (2023c). E-mail Spam Detection Using Machine Learning. *2023 4th International Conference for Emerging Technology, INCET 2023*, *11*(5), 350–356. https://doi.org/10.1109/INCET57972.2023.10170187

Statista - The Statistics Portal for Market Data, Market Research and Market Studies. (n.d.). Retrieved September 1, 2024, from https://www.statista.com/

Suryawanshi, S., Goswami, A., & Patil, P. (2019). Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers. *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 69–74. https://doi.org/10.1109/IACC48062.2019.8971582

Sweet, L., Müller, C., Anand, M., & Zscheischler, J. (2023). Cross-Validation Strategy Impacts the Performance and Interpretation of Machine Learning Models. *Artificial Intelligence for the Earth Systems*, *2*(4), e230026. https://doi.org/10.1175/AIES-D-23-0026.1

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 252–259.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Verma, T. (2017). Impact factor: 4.295 E-Mail Spam Detection and Classification Using SVM and Feature Extraction. *International Journal of Advance Research*, *3*, 1491–1495. www.ijariit.com

Wankhade, N. V., Keole, R. R., & Mahore, T. R. (2022). *Review paper on Spam Email Detection with Classification Using Machine Learning*. *9*(8), 1–6.

Wilimitis, D., & Walsh, C. G. (2023). Practical Considerations and Applied Examples of Cross-Validation for Model Development and Evaluation in Health Care: Tutorial. *JMIR AI*, *2*, e49023. https://doi.org/10.2196/49023

Yue Kang Zhao Cai, C.-W. T. Q. H., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, *7*(2), 139–172. https://doi.org/10.1080/23270012.2020.1756939

Zavrak, S., & Yilmaz, S. (2023). Email spam detection using hierarchical attention hybrid deep learning method. *Expert Systems with Applications*, *233*, 120977. https://doi.org/10.1016/j.eswa.2023.120977

Zhou, B., Yao, Y., & Luo, J. (2010). *A Three-Way Decision Approach to Email Spam Filtering BT - Advances in Artificial Intelligence* (A. Farzindar & V. Kešelj (Eds.); pp. 28–39). Springer Berlin Heidelberg.

Appendixes

Appendix 1: Installing and Importing Relevant Libraries

```
pip install pandas googletrans==4.0.0-rc1 langdetect

pip install deep-translator

from googletrans import Translator
from langdetect import detect

import pandas as pd

import re
import nltk
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, validation_curve
from imblearn.over_sampling import SMOTE
import numpy as np
import matplotlib.pyplot as plt
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score, KFold
from sklearn.metrics import classification_report, accuracy_score, pre
cision_recall_fscore_support
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

from sklearn.naive_bayes import MultinomialNB

from sklearn.naive_bayes import BernoulliNB

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier
```

Appendix 2: Split the Dataset into Chunks

```
def split_csv(file_path, chunk_size, output_prefix):
        # Read the CSV file in chunks with the specified encoding
        for i, chunk in enumerate(pd.read_csv(file_path,
    chunksize=chunk_size, encoding='ISO-8859-1')):
            # Save each chunk to a new CSV file
            chunk.to_csv(f"{output_prefix}_{i + 1}.csv", index=False)
            print(f"Chunk {i + 1} saved!")
```

```
file_path = "k_data.csv"
chunk_size = 25  # Number of rows per chunk
output_prefix = "chunk"
```

Appendix 3: Data Translation

```
# Initialize the translator
translator = Translator()

# Function to translate a single text
def translate_text(text, target_language='am'):
    try:
        detected_language = detect(text)
        print(f"Detected language: {detected_language}")
        translated = translator.translate(text, src=detected_language, de
st=target_language)
        return translated.text
    except Exception as e:
        print(f"Error translating text: {text[:50]}... Error: {e}")
        return text

# Combine chunks, translate, and save as one file
def translate_and_save(csv_files, output_file):
    translated_chunks = []
    for file in csv_files:
        print(f"Processing {file}...")
        chunk = pd.read_csv(file)
        chunk['translated'] = chunk['subject'].apply(lambda x: translate_
text(x))  # Replace 'text_column' with your column name
        translated_chunks.append(chunk)

    # Combine all translated chunks into one DataFrame
    combined_df = pd.concat(translated_chunks)
    combined_df.to_csv(output_file, index=False)
    print(f"Translated data saved to {output_file}")

# List of CSV files to process
csv_files = [f"chunk_{i}.csv" for i in range(1, 223)]  # Add all your chu
nk file names here
output_file = "k_merged_data.csv"

# Translate and save
translate_and_save(csv_files, output_file)
```

Appendix 4: Amharic Data Cleaning

```
# Load the dataset
df = pd.read_csv('emailSubject_amharic.csv')

# Step 1: Remove duplicates
df.drop_duplicates(inplace=True)

# Step 2: Handle missing values
# Remove rows with missing email content
df.dropna(subset=['subject'], inplace=True)

# Step 3: Normalize text
def normalize_amharic_text(text):
    text = text.lower()  # Convert to lowercase (Note: Amharic may not ha
ve case differences, but this is a safeguard)
    text = re.sub(r'\d+', '', text)  # Remove numbers
    text = re.sub(r'\s+', ' ', text)  # Remove extra whitespace
    text = re.sub(r'[^\w\s፡።፣፤፥፦፧፨]', '', text)  # Remove punctuation a
nd special characters, keep Amharic-specific punctuation
    return text

df['email'] = df['email'].apply(normalize_amharic_text)

# Step 4: Tokenization
def tokenize_amharic_text(text):
    words = word_tokenize(text)
    return ' '.join(words)

df['subject'] = df['subject'].apply(tokenize_amharic_text)

# Load Amharic stopwords from CSV
stopwords_df = pd.read_csv('stopwords-am.csv', header=None)
amharic_stop_words = stopwords_df[0].tolist()

# Step 4: Tokenization and Stopwords Removal
def remove_amharic_stopwords(text):
    words = word_tokenize(text)
    filtered_words = [word for word in words if word not in amharic_stop_
words]
    return ' '.join(filtered_words)

df['subject'] = df['subject'].apply(remove_amharic_stopwords)

# Save the cleaned data to a new CSV file
df.to_csv('cleaned_amharic_emails.csv', index=False)
```

Appendix 5: Feature Extraction from Combined dataset using TF-IDF

```
# Load your datasets
amharic_df = pd.read_csv("cleaned_amharic_emails.csv")

# TF-IDF Vectorization
tfidf = TfidfVectorizer(max_features=5000)
X = tfidf.fit_transform(amharic_df ['subject'])
y = amharic_df ['label']
```

Appendix 6: MNB Model Training

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resample
d, test_size=0.3, random_state=42)

# Train a model using MultinomialNB
model = MultinomialNB()
model.fit(X_train, y_train)

MultinomialNB()

# Make predictions
y_pred = model.predict(X_test)

# Get precision, recall, f1-score for each class
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average=None)

# Calculate the mean of precision, recall, and f1-score across both classes
mean_precision = precision.mean()
mean_recall = recall.mean()
mean_f1_score = f1.mean()

# Output the results
print(f'Mean Precision: {mean_precision:.4f}')
print(f'Mean Recall: {mean_recall:.4f}')
print(f'Mean F1-Score: {mean_f1_score:.4f}')
```

Appendix 7: BNB Model Training

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resample
d, test_size=0.3, random_state=42)

# Train a model using BernoulliNB
model = BernoulliNB()
model.fit(X_train, y_train)
```

```
# Make predictions
y_pred = model.predict(X_test)

# Get precision, recall, f1-score for each class
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average=None)

# Calculate the mean of precision, recall, and f1-score across both classes
mean_precision = precision.mean()
mean_recall = recall.mean()
mean_f1_score = f1.mean()

# Output the results
print(f'Mean Precision: {mean_precision:.4f}')
print(f'Mean Recall: {mean_recall:.4f}')
print(f'Mean F1-Score: {mean_f1_score:.4f}')
```

Appendix 8: SVM Model Training

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resample
d, test_size=0.3, random_state=42)

# Train a model using SVM
model = SVC()
model.fit(X_train, y_train)

SVC()

# Make predictions
y_pred = model.predict(X_test)

# Get precision, recall, f1-score for each class
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average=None)

# Calculate the mean of precision, recall, and f1-score across both classes
mean_precision = precision.mean()
mean_recall = recall.mean()
mean_f1_score = f1.mean()

# Output the results
print(f'Mean Precision: {mean_precision:.4f}')
print(f'Mean Recall: {mean_recall:.4f}')
print(f'Mean F1-Score: {mean_f1_score:.4f}')
```

Appendix 9: RF Model Training

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resample
d, test_size=0.3, random_state=42)

# Initialize the Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

RandomForestClassifier(random_state=42)

# Make predictions
y_pred = model.predict(X_test)

# Get precision, recall, f1-score for each class
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average=None)

# Calculate the mean of precision, recall, and f1-score across both classes
mean_precision = precision.mean()
mean_recall = recall.mean()
mean_f1_score = f1.mean()

# Output the results
print(f'Mean Precision: {mean_precision:.4f}')
print(f'Mean Recall: {mean_recall:.4f}')
print(f'Mean F1-Score: {mean_f1_score:.4f}')
```