Deep Learning Approach For Digital Forensics Analysis In Electronic Mails

A PROJECT REPORT

Submitted by

KRITHIKA S (810020205042)

LAVANYA M (810020205044)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING

BIT CAMPUS, TIRUCHIRAPPALLI

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2024

UNIVERSITY COLLEGE OF ENGINEERING
BIT CAMPUS
TIRUCHIRAPPALLI-620 024

BONAFIDE CERTIFICATE

Certified that this project report "**Deep Learning Approach For Digital Forensics Analysis In Electronic Mails"** is the bonafide work of "**Ms. KRITHIKA S (810020205042)** and **Ms. LAVANYA M (810020205044)"** who carried out the project work under my supervision.

SIGNATURE
Dr. G. ANNAPOORANI
HEAD OF THE DEPARTMENT
Assistant Professor
Department of IT/CSE
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

SIGNATURE
Dr. M. PADMA
SUPERVISOR
Teaching Fellow
Department of CSE
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

Submitted for the project Viva - voce examination held on …….....

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that the work entitled "**Deep Learning Approach For Digital Forensics Analysis In Electronic Mails**" is submitted in partial fulfillment of the requirement for the award of the degree in B. Tech, in University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. It is the record of our own work carried out during the academic year 2023 – 2024 under the supervision and guidance of **Dr. M. PADMA**, Teaching Fellow, Department of CSE, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places.

Krithika S (810020205042)

Lavanya M (810020205044)

I certify that the declaration made above by the candidates is true

Signature of the Guide

Dr. M. PADMA,
Teaching Fellow,
Department of CSE,
University College of Engineering,
Anna University – BIT Campus,
Tiruchirappalli – 620 024.

ACKNOWLEDGEMENT

We would like to thank our honourable Dean **Dr. T. SENTHIL KUMAR**, Professor for having provided us with all required facilities to complete our project without hurdles.

We would also like to express our sincere thanks to **Dr. G. ANNAPOORANI**, Head of the Department, Department of Computer Science and Engineering, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

We would like to thank our Project Coordinator **Dr. S. USHA**, Assistant Professor, Department of Computer Science and Engineering, **Mrs. M. REVATHI**, Teaching Fellow, Department of Computer Science and Engineering for there kind support.

We would like to thank and express our deep sense of gratitude to our project guide **Dr. M. PADMA,** Teaching Fellow, Department of Computer Science and Engineering, for her valuable guidance throughout the project.

We also extend our thanks to all other teaching and non-teaching staff for their encouragement and support.

We thank our beloved parents and friends for their full support in the moral development of this project.

ABSTRACT

E-mail is an essential application for carrying out transactions and efficiency in business processes to improve productivity. E-mail is frequently used as a vital medium of communication and is also being used by cybercriminals to commit crimes. Cybercrimes like hacking, spoofing, phishing, E-mail bombing, whaling, and spamming are being performed through E-mails. Hence, there is a need for proactive data analysis to prevent cyber-attacks and crimes. To investigate crimes involving Electronic Mail (e-mail), analysis of both the header and the email body is required since the semantics of communication helps to identify the source of potential evidence. With the continued growth of data shared via emails, investigators now face the daunting challenge of extracting the required semantic information from the bulks of emails, thereby causing a delay in the investigation process. The existing email classification approaches lead towards irrelevant E-mails and/or loss of valuable information. Keeping in sight these limitations, our project proposed to design a novel efficient approach for E-mail classification into four different classes: Normal, Fraudulent, Harassment, and Suspicious E-mails by using LSTM and GRU. The LSTM and GRU efficiently captures meaningful information from E-mails that can be used for forensic analysis as evidence. It effectively outperforms existing methods while keeping the classification process robust and reliable.

Keywords: E-mail classification, LSTM, GRU, Cybercrimes, Phishing, Spoofing, Email bombing, Semantic information.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
DL	–	Deep Learning
ML	–	Machine Learning
CNN	–	Convolutional Neural Network
Conv2D	–	Convolutional 2Dimensional
OpenCV	–	Open Computer Vision.
ReLU	–	Rectified Linear Unit
RNN	–	Recurrent Neural Network
FNN	-	Feedforward Neural Network
FSM	-	Finite State Machine
HCI	-	Human Computer Interaction
STT	-	Speech-To-Text
TTS	-	Text-To-Speech
LSTM	-	Long Short-Term Memory
API	-	Application Program Interface
COM	-	Component Object Model
GUI	-	Graphical User Interfac

CHAPTER 1

INTRODUCTION

1.2 INTRODUCATION

In the digital era, Electronic Mail, commonly known as email, stands as a cornerstone of communication, facilitating the exchange of messages across vast distances with ease. Initially conceived as a means of electronic correspondence between computers, email has evolved into a ubiquitous tool utilized in various domains, including business, education, and technical communication. Its versatility allows individuals to interact seamlessly, transcending geographical boundaries and time constraints. The inception of email dates back to 1971 when Ray Tomlinson sent the first test message to himself, marking the dawn of electronic communication. Since then, email has undergone significant transformations, becoming an integral part of everyday life. Messages are transmitted through email servers, leveraging multiple protocols within the TCP/IP suite. Notably, SMTP (Simple Mail Transfer Protocol) facilitates message sending, while protocols like IMAP (Internet Message Access Protocol) or POP (Post Office Protocol) enable message retrieval from mail servers. Accessing email accounts has become increasingly streamlined, with most webmail services automating the configuration process. However, manual configuration may be necessary when using email clients such as Microsoft Outlook or Apple Mail, requiring users to input server details and port numbers alongside their credentials. Email messages comprise three fundamental components: the message envelope, depicting the email's electronic format; the message header, containing essential metadata such as subject line and sender/recipient information; and the message body, encompassing textual content, images, and file attachments. While email facilitates seamless communication and enhances

productivity in various domains, its pervasive usage has also attracted malicious actors. Cybercriminals exploit email channels for nefarious activities such as hacking, spoofing, phishing, email bombing, whaling, and spamming. As a result, there is an imperative for proactive data analysis to mitigate cyber threats and safeguard digital assets. Forensic investigation involving electronic mail necessitates comprehensive analysis of both email headers and bodies. Semantic insights derived from communication patterns aid in identifying potential evidence sources. However, the burgeoning volume of email data poses challenges for investigators, necessitating efficient techniques for extracting semantic information without compromising investigation timelines. Existing email classification methods often fall short, leading to misclassification or information loss. In response, our project endeavors to devise an innovative approach leveraging deep learning techniques, specifically Long Short-Term Memory (LSTM) based Gated Recurrent Unit (GRU), for email classification. By categorizing emails into four distinct classes - Normal, Fraudulent, Harassment, and Suspicious - our approach aims to enhance forensic analysis by capturing meaningful information from email contents. Through rigorous evaluation, we demonstrate the superiority of our method in terms of robustness and reliability, thereby addressing the limitations of existing approaches in email classification. Within the domain of deep learning, our project contributes to advancing email classification methodologies, offering promising avenues for combating cybercrimes and safeguarding digital communications.

1.2 DOMAIN

Deep learning is a subset of artificial intelligence (AI) that has gained significant traction in recent years due to its remarkable capabilities in solving complex problems across various domains. It involves training artificial neural networks, comprising multiple layers of interconnected nodes, to learn patterns and representations from data. Unlike traditional machine learning techniques

that rely on feature engineering, deep learning algorithms automatically extract hierarchical features from raw data, making them particularly adept at handling unstructured and high-dimensional data such as images, text, and audio. The success of deep learning can be attributed to several factors, including the availability of vast amounts of data, advancements in computational hardware (e.g., GPUs), and breakthroughs in neural network architectures. Convolutional Neural Networks (CNNs) have revolutionized image recognition tasks, while Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), excel in sequential data processing tasks like natural language processing (NLP) and time series prediction. Deep learning finds applications across diverse domains, including computer vision, speech recognition, natural language understanding, healthcare, finance, autonomous vehicles, and more. In computer vision, deep learning models power object detection, image classification, and segmentation tasks with unprecedented accuracy. In NLP, they enable language translation, sentiment analysis, text summarization, and chatbots capable of engaging in human-like conversations. The flexibility and scalability of deep learning frameworks such as TensorFlow, PyTorch, and Keras have democratized access to deep learning tools, fostering innovation and accelerating research in academia and industry. Transfer learning techniques allow practitioners to leverage pre-trained models and adapt them to new tasks with limited data, further lowering the barrier to entry for deploying deep learning solutions. Despite its remarkable successes, deep learning is not without challenges. Training deep neural networks often requires large amounts of labeled data and substantial computational resources, making it resource-intensive and computationally expensive. Additionally, deep learning models can be susceptible to adversarial attacks, where small perturbations to input data lead to incorrect predictions. In conclusion, deep learning represents a paradigm shift in AI, enabling machines to learn complex patterns and representations directly from data. With its versatility and ability to

tackle a wide range of problems, deep learning continues to push the boundaries of what is possible in AI-driven solutions, shaping the future of technology and society.

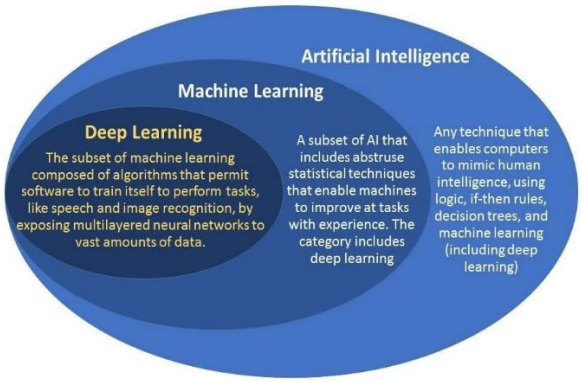


Fig 1.2.1 Artificial intelligence vs. Machine learning vs. Deep learning

1.3 PROBLEM STATEMENT

Email is a universal service used by over a billion people worldwide. As one of the most popular services, email has become a major vulnerability to users and organizations. The statistics are astounding. Email remains the number one threat vector for data breaches, the point of entry for ninety-four percent of breaches. There is an attack every 39 seconds. Over 30% of phishing messages get opened, and 12% of users click on malicious links. As cybercrime becomes more advanced and bypasses the legacy controls put in place to defend against it, security must become more advanced too. Below are some of the most common types of Attacks:

1.3.1 Phishing

Phishing is a form of fraud. Cyber criminals use email, instant messaging, or other social media to try to gather information such as login credentials by masquerading as a reputable person. Phishing occurs when a malicious party sends a fraudulent email disguised as being from an authorized, trusted source. The message intent is to trick the recipient into installing malware on his or her

device or into sharing personal or financial information.Spear phishing is a highly targeted phishing attack. While phishing and spear phishing both use emails to reach the victims, spear-phishing sends customized emails to a specific person. The criminal researches the target's interests before sending the email.

1.3.2 Vishing

Vishing is phishing using voice communication technology. Criminals can spoof calls from authorized sources using voice-over IP technology. Victims may also receive a recorded message that appears authorized. Criminals want to obtain credit card numbers or other information to steal the victim's identity. Vishing takes advantage of the fact that people trust the telephone network.

1.3.3 Smishing

Smishing is phishing using text messaging on mobile phones. Criminals impersonate a legitimate source in an attempt to gain the trust of the victim. For example, a smishing attack might send the victim a website link. When the victim visits the website, malware is installed on the mobile phone.

1.3.4 Whaling

Whaling is a phishing attack that targets high profile targets within an organization such as senior executives. Additional targets include politicians or celebrities.

1.3.5 Pharming

Pharming is the impersonation of an authorized website in an effort to deceive users into entering their credentials. Pharming misdirects users to a fake website that appears to be official. Victims then enter their personal information thinking that they are connected to a legitimate site.

1.3.6 Spyware

Spyware is software that enables a criminal to obtain information about a

user's computer activities. Spyware often includes activity trackers, keystroke collection, and data capture. In an attempt to overcome security measures, spyware often modifies security settings. Spyware often bundles itself with legitimate software or with Trojan horses. Many shareware websites are full of spyware.

1.3.7 Scareware

Scareware persuades the user to take a specific action based on fear. Scareware forges pop-up windows that resemble operating system dialogue windows. These windows convey forged messages stating that the system is at risk or needs the execution of a specific program to return to normal operation. In reality, no problems exist, and if the user agrees and allows the mentioned program to execute, malware infects his or her system.

1.3.8 Adware

Adware typically displays annoying pop-ups to generate revenue for its authors.The malware may analyse user interests by tracking the websites visited. It can then send pop up advertising relevant to those sites. Some versions of software automatically install Adware.

1.3.9 Spam

Spam (also known as junk mail) is unsolicited email. In most cases, spam is a method of advertising. However, spam can send harmful links, malware, or deceptive content. The end goal is to obtain sensitive information such as a social security number or bank account information. Most spam comes from multiple computers on networks infected by a virus or worm. These compromised computers send out as much bulk email as possible.

1.3.10 E-Mail Bombing

An email bombing is an attack on your inbox that involves sending massive amounts of messages to your address. Sometimes these messages are complete

gibberish, but more often they'll be confirmation emails for newsletters and subscriptions. In the latter case, the attacker uses a script to search the internet for forums and newsletters and then signs up for an account with your email address. Each will send you a confirmation email asking to confirm your address. This process repeats across as many unprotected sites as the script can find.

1.4 OBJECTIVES

- **Utilizing LSTM and GRU neural networks for threat detection:** This objective outlines the use of deep learning techniques to detect various email-based cyber threats like harassment messages, fraudulent messages, and suspicious content.
- **Creating a digital forensic analysis system for email communications:** This objective focuses on developing a system to analyze email communications, identify evidence of cybercrimes, attribute malicious activities, and provide insights for law enforcement purposes.
- **Implementing automated anomaly identification and classification:** This objective aims to automate the identification and classification of anomalous emails, enabling proactive responses such as alerts, deletion of malicious emails, and further investigation procedures to mitigate risks.

1.5 OUTLINE

This project document is organized as follows.

- In chapter 2, the literature review is presented
- In chapter 3, System study
- In chapter 4, Proposed system
- In chapter 5, System requirements and Software specifications
- In chapter 6, System design
- In chapter 7, Implementation and Results
- Finally, In chapter 8, Conclusion and future enhancement are drawn

CHAPTER 2

LITERATURE SURVEY

[1] (ASIF KARIM, (Member, IEEE), SAMI AZAM et al.,2021) The authors aim to devise an unsupervised framework for email classification, focusing on clustering methodologies. By leveraging both email content (body) and subject headers, the framework employs ten distinct features, with seven specifically tailored for this study. Through extensive experimentation with various clustering algorithms, including OPTICS and DBSCAN, the study achieves a notable 0.26% increase in average efficiency with OPTICS outperforming others. The approach showcases a promising average balanced accuracy of 75.76%, emphasizing its potential for robust spam and ham email classification.

[2] (Manoj Sethi, Sumesha Chandra, Vinayak Chaudhary, Yash,2021) The Spam emails are known as unrequested commercialized emails or deceptive emails sent to a specific person or a company. Spams can be detected through natural language processing and machine learning methodologies. Machine learning methods are commonly used in spam filtering. These methods are used to render spam classifying emails to either ham (valid messages) or spam (unwanted messages) with the use of Machine Learning classifiers. The proposed work differentiating features of the content of documents. There has been a lot of work that has been performed in the area of spam filtering which is limited to some domains. Research on spam email detection either focuses on natural language processing methodologies on single machine learning algorithms or one natural language processing technique on multiple machine learning algorithms. A modeling pipeline is developed to review the machine learning methodologies. The future work includes testing the model with various standard datasets. The

research proposes that the outcome that is obtained should be compared with additional spam datasets from various sources. Also, more classification and feature algorithms should be analyzed with email spam datasets.

[3] (P.U. Anitha, Dr.C.V. Guru Rao, Dr. D. Suresh Babu, 2021) The authors aim to proposed an E-mail spam, known as undesirable Bulk E-mail (UBE), junk mail, or undesirable commercial e-mail (UCE), is transferring undesirable e-mail information, usually with business data, in large amounts to a confused set of recipients. Spam is standard on the Internet because automated communications' transaction costs are lower than other alternative forms of communication. Many spam filters use various approaches to recognize the incoming message as spam, varying from white list/blacklist, Bayesian review, keyword matching, postage, mail header analysis, enactment, etc. Even though we are still involved in spam e-mails every day. It proposes an enhanced spam exposure design based on Extreme Gradient Boosting (XGBoost) model. It is studied for increased accuracy in spam detection. In this paperss XGBoot classifier to detect the spam emails from that given dataset. The proposed XGBoost classifier used as a spam emails indicator and utilizes a standard computational intellect method on a benchmark dataset. The proposed method compared with current classifiers of SVM, CNSA-FFO, Rotation forest, MLP, J48, and Naïve Bayes classifiers. The evaluation results confirm that the proposed model got better accuracy with 95% compared with the current approaches.

[4] (Zeeshan Bin Siddique, Mudassar Ali Khan et al,2021) The authors aim to proposed an E-mail services have been evolved into a powerful tool for the exchange of different kind of information. The increased use of the e-mail also entails more spam attacks for the Internet users. Spam can be sent from anywhere on the planet from users having deceptive intentions that has access to the Internet. These spam emails have fake content with mostly links for phishing attacks and other threats, and these emails are sent in bulk to a large number of

recipients. It describes how machine learning (ML) and deep learning (DL) models such as Support Vector Machine (SVM), Naïve Bayes, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), a recurrent neural network, can be trained to detect spam emails. Accuracy, precision, recall, F1-measure, ROC-AUC, and model loss are used as comparative measures to examine performance. The study concludes that deep learning models are more successful in classifying spam emails. Comparatively, LSTM algorithm has a high accuracy rate of around 98% with low model loss rate of 5%. Even though LSTM takes a little longer to train than CNN, SVM, or Naïve Bayes, its efficiency and accuracy rate are far better than those of the other approaches.

[5] (Ganiev Salim Karimovich, Khamidov Sherzod Jaloldin ugli et al,2021) The authors aim to proposed an Electronic-mail (abbreviated as e-mail) is a fast, effective and expensive method of exchanging messages over the Internet. Clicking on links in a spam email can direct users to phishing websites or places that are infected with malicious software. Spam being a carrier of malware causes the proliferation of unsolicited advertisements, fraud schemes, phishing messages, explicit content, promotions of cause, etc. It describes the advantages and disadvantages of methods for detecting and protecting against spam messages in electronic mail services, considers the classification of spam messages and the Naive Bayes spam filter used to classify spam messages. The characteristics and effectiveness of spam filters based on machine learning methods are analyzed. Machine learning algorithms play a central role in detection of spam e-mail. In this paper, the author presented an empirical evaluation of three machine learning algorithms for spam filtering. These approaches, NB, k-NN, and SVM, were applied to different parts of an e-mail in order to compare their performance.

[6] (Nikhil Kumar, Sanket Sonowal, Nishant, 2020) The authors aim to proposed an Email Spam has become a major problem nowadays, with Rapid

growth of internet users, Email spams is also increasing. Creating a fake profile and email account is much easy for the spammers, they pretend like a genuine person in their spam emails, these spammers target those peoples who are not aware about these frauds. It will discuss the machine learning algorithms and apply all these algorithms on our data sets and best algorithm is selected for the email spam detection having best precision and accuracy. Automatic email filtering may be the most effective method of detecting spam but nowadays spammers can easily bypass all these spam filtering applications easily. Most of the spam can be blocked manually coming from certain email addresses. Machine learning approach will be used for spam detection. It can be concluded that the Multinomial Naïve Bayes gives the best outcome but has limitation due to class-conditional independence which makes the machine to misclassify some tuples. "Filtering of spams can be done on the basis of the trusted and verified domain names." "The spam email classification is very significant in categorizing e-mails and to distinct e-mails that are spam or non-spam." "It can be used by the big body to differentiate decent mails that are only the emails they wish to obtain."

[9] (Vinodhini. M, Prithvi. D, Balaji. S,2020) The authors aim to proposed an current use of social media has created incomparable amounts of social data, as it is a cheap and popular information sharing communication platform. Identifying the spammers and spam material is a hot subject of study, and while large amounts of experiments have recently been conducted to this end, so far the methodologies are only barely able to identify spam feedback, and none of them demonstrates the value of each derived function type. In this study, a machine learning-based spam detection system that determines whether or not a specific message in the dataset is spam using a set of machine learning algorithms. Four main features have been used; including user behavioural, user-linguistic, review behavioural and review-linguistic, to improve the spam detection process and to gather reliable data. In this paper, the spams are identified and spammers present

in a twitter dataset with the help of machine learning algorithms and NLP concepts. The user behavior is determined with the help of properties such as reviews written and an average of negative ratio given. Thus, making it a very effective and accurate spam detection framework.

CHAPTER 3

SYSTEM STUDY

3.1 EXISTING SYSTEM

Aims to devise an unsupervised framework for email classification, focusing on clustering methodologies. By leveraging both email content (body) and subject headers, the framework employs ten distinct features, with seven specifically tailored for this study. Through extensive experimentation with various clustering algorithms, including OPTICS and DBSCAN, the study achieves a notable 0.26% increase in average efficiency with OPTICS outperforming others. The approach showcases a promising average balanced accuracy of 75.76%, emphasizing its potential for robust spam and ham email classification.

DISADVANTAGE

- The unsupervised approach may struggle to adapt to emerging spam techniques or variations in email content used by sophisticated attackers, potentially leading to misclassification or overlooking of new spam patterns.
- Effective clustering in email data hinges on well-crafted features, as poor selection can skew results. Large-scale processing of emails for clustering faces scalability hurdles, straining computational resources and time. This limits its utility in real-time or high-throughput email settings.
- While the clustering approach may effectively classify emails into spam and ham categories, the black-box nature of some clustering algorithms may hinder the interpretability of the results, making it challenging to understand the underlying patterns or reasons behind the classification decisions.

CHAPTER 4

PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

Our proposed system is to classify emails into four distinct classes: Normal, Fraudulent, Harassment, and Suspicious. By leveraging advanced deep learning techniques such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), our system seeks to enhance the efficiency and accuracy of email classification, thereby facilitating proactive data analysis to prevent cyber-attacks and crimes.

4.2 LSTM AND GRU

In Deep learning, Long-Term Short-Term Memory Networks and Gated Recurrent Units, LSTM and GRUs for short.

4.2.1 LSTM – Long Short-Term Memory

LSTM are a special kind of RNN which is capable of learning long-term dependencies. LSTMs are designed to dodge long-term dependency problem as they are capable of remembering information for longer periods of time. Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the

cell. The expression long short-term refers to the fact that LSTM is a model for the short-term memory which can last for a long period of time. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs. The popularity of LSTM is due to the Getting mechanism involved with each LSTM cell. In a normal RNN cell, the input at the time stamp and hidden state from the previous time step is passed through the activation layer to obtain a new state. Whereas in LSTM the process 6 is slightly complex, as you can see in the above architecture at each time it takes input from three different states like the current input state, the short-term memory from the previous cell and lastly the long-term memory.

Initialization:

The LSTM layer is initialized with parameters such as the number of LSTM units, dropout rates, and recurrent dropout rates.

Forward Pass:

During the forward pass, the LSTM layer processes the input sequences (email text) token by token, updating its internal state at each time step.

Internal Computations:

At each time step, LSTM units compute various gates and activation functions to update their internal state, including forget gate, input gate, candidate cell state, and output gate.

Backpropagation:

During training, backpropagation through time (BPTT) is performed to compute gradients and update the LSTM parameters (weights and biases) based on the loss between predicted and actual outputs.

Formulas Used in LSTM:

Forget Gate:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C_t} = tanh(W_i . [h_{t-1}, x_t] + b_c$$

Update Cell State:

$$C_t = f_t . C_{t-1} + i_t . \widetilde{C_t}$$

Output Gate:

$$\boldsymbol{o_t} = \sigma(W_o \cdot [h_{t-1}, x_t] + b_0)$$

$$h_t = o_{\boldsymbol{t}} \cdot \tanh(C_t)$$

4.2.2 GRU – Gated Recurrent Unit

Gated recurrent unit (GRU) was introduced by Cho, et al. in 2014 to solve the vanishing gradient problem faced by standard recurrent neural networks (RNN). GRU shares many properties of long short-term memory (LSTM). Both algorithms use a gating mechanism to control the memorization process. A gated recurrent unit (GRU) is a gating mechanism in recurrent neural networks (RNN) similar to a long short-term memory (LSTM) unit but without an output gate. GRU's try to solve the vanishing gradient problem that can come with standard recurrent neural networks. A GRU can be considered a variation of the long short-term memory (LSTM) unit because both have a similar design and produce equal results in some cases. GRU's are able to solve the vanishing gradient problem by using an update gate and a reset gate. The update gate controls information that flows into memory, and the reset gate controls the information that flows out of memory. The update gate and reset gate are two vectors that decide which

information will get passed on to the output. They can be trained to keep information from the past or remove information that is irrelevant to the prediction. A GRU is a very useful mechanism for fixing the vanishing gradient problem in recurrent neural networks. The vanishing gradient problem occurs in machine learning when the gradient becomes vanishingly small, which prevents the weight from changing its value. They also have better performance than LSTM when dealing with smaller datasets. The workflow of the Gated Recurrent Unit, in short GRU, is the same as the RNN but the difference is in the operation and gates associated with each GRU unit. To solve the problem faced by standard RNN, GRU incorporates the two gate operating mechanisms called Update gate and Reset gate.

Initialization:

Similar to LSTM, the GRU layer is initialized with parameters such as the number of GRU units, dropout rates, and recurrent dropout rates.

Forward Pass:

During the forward pass, the GRU layer processes the input sequences (email text) token by token, updating its internal state at each time step.

Internal Computations:

At each time step, GRU units compute reset gate, update gate, and candidate hidden state to update their internal state.

Backpropagation:

Similar to LSTM, backpropagation through time (BPTT) is performed during training to update the GRU parameters based on the loss between predicted and actual outputs.

Formulas Used in GRU:

Reset Gate:

$$r_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_r)$$

Update Gate:

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t] + b_z)$$

Candidate Hidden State:

$$\widetilde{h_t} = tanh(W_h . [r_t . h_{t-1}, x_t] + b_h$$

Hidden State Update:

$$h_t = (1 - z_t) . h_{t-1} + z_t . \widetilde{h_t}$$

Dense Layer:

Output layer with softmax activation for multi-class classification.

4.3 ADVANTAGE OF PROPOSED SYSTEM

Our model categorize emails into four distinct classes: Normal, Harassing, Suspicious and Fraudulent offering more precise classification than binary approaches and use of LSTM and GRU architecture leverages deep learning's power to capture long-term dependencies and hidden pattern in email text, potentially leading to improved accuracy.

CHAPTER 5

SYSTEM REQUIREMENTS AND SOFTWARE SPECIFICATIONS

5.1 HARDWARE REQUIREMENTS

System : AMD PRO A4-4350B R4, 5 COMPUTE CORES 2C+3G 2.50 GHz

Hard disk : 1 TB

RAM : 4 GB

5.2 SOFTWARE REQUIREMENTS

Operating System : WINDOWS 10

Language Used : PYTHON

5.3 LIBRARIES USED

- PYTHON
- NUMPY
- PANDAS
- BEAUTIFULSOUP (BS4)
- SCIKIT-LEARN
- TENSORFLOW
- EMAIL
- SMTPLIB
- IMAPLIB
- IMAPCLIENT

PYTHON

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its

notable use of significant indentation. It's language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale project's python is dynamically-typed and garbage-collected. It supports multiple programming, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to it's comprehensive standard library. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

NUMPY

A powerful library for numerical computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

PANDAS

A versatile data manipulation and analysis library that offers data structures like DataFrame and Series, making it easy to work with structured data, perform data cleaning, manipulation, and analysis tasks.

BEAUTIFULSOUP (BS4)

A Python library for pulling data out of HTML and XML files. It provides simple methods and Pythonic idioms for navigating, searching, and modifying parsed data, making it useful for web scraping tasks.

SCIKIT-LEARN

A comprehensive machine learning library built on NumPy, SciPy, and matplotlib. It provides simple and efficient tools for data mining and data

analysis, including various algorithms for classification, regression, clustering, dimensionality reduction, and model evaluation.

TENSORFLOW

An open-source machine learning framework developed by Google for building and training deep learning models. It provides a flexible ecosystem of tools, libraries, and community resources to support research and production-level deployment of machine learning models.

EMAIL

A built-in Python library for handling email messages, including parsing, composing, and sending emails. It provides functions and classes to work with email messages and related tasks.

SMTPLIB

A built-in Python library that defines SMTP client sessions to send emails to any Internet machine with an SMTP or ESMTP listener daemon.

IMAPLIB

A built-in Python library for accessing and manipulating mailboxes on a remote IMAP server. It allows you to perform various operations such as searching, fetching, and deleting emails from the server.

IMAPCLIENT

A third-party Python library that provides an easy-to-use, Pythonic interface for working with IMAP mail servers. It simplifies the process of interacting with IMAP servers and handling emails.

CHAPTER 6

SYSTEM DESIGN

6.1 UML DIAGRAM

Unified Modeling Language (UML) diagrams are a standardized visual representation used in software engineering to model systems, describe their structure, behavior, and interactions. UML provides a common language and notation that allows developers, stakeholders, and other involved parties to communicate and understand the various aspects of a system's design and functionality.

6.1.1 USECASE DIAGRAM

This diagram depicts the interaction between the email user and the email classification system. The email classification system utilizes advanced deep learning techniques such as LSTM and GRU to classify emails into four distinct classes: Normal, Fraudulent, Harassment, and Suspicious.

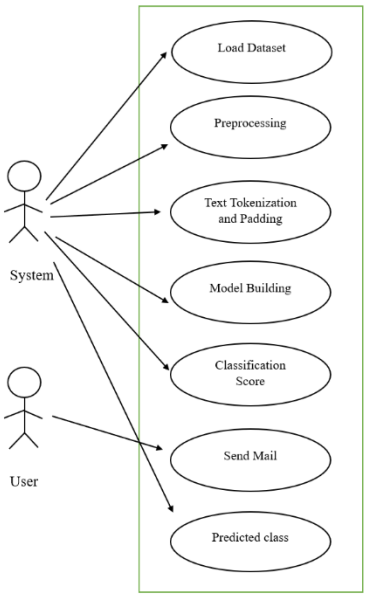


Fig 6.1.1.1 Use Case Diagram

6.1.2 DATA FLOW DIAGRAM

6.1.2.1 DFD LEVEL 0

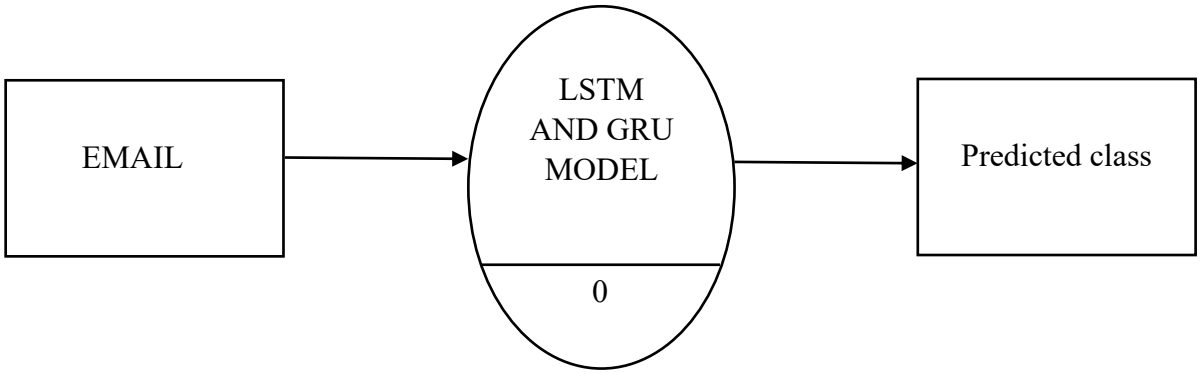


Fig 6.1.2.1.1 DFD Level 0

6.1.2.2 DFD LEVEL 1

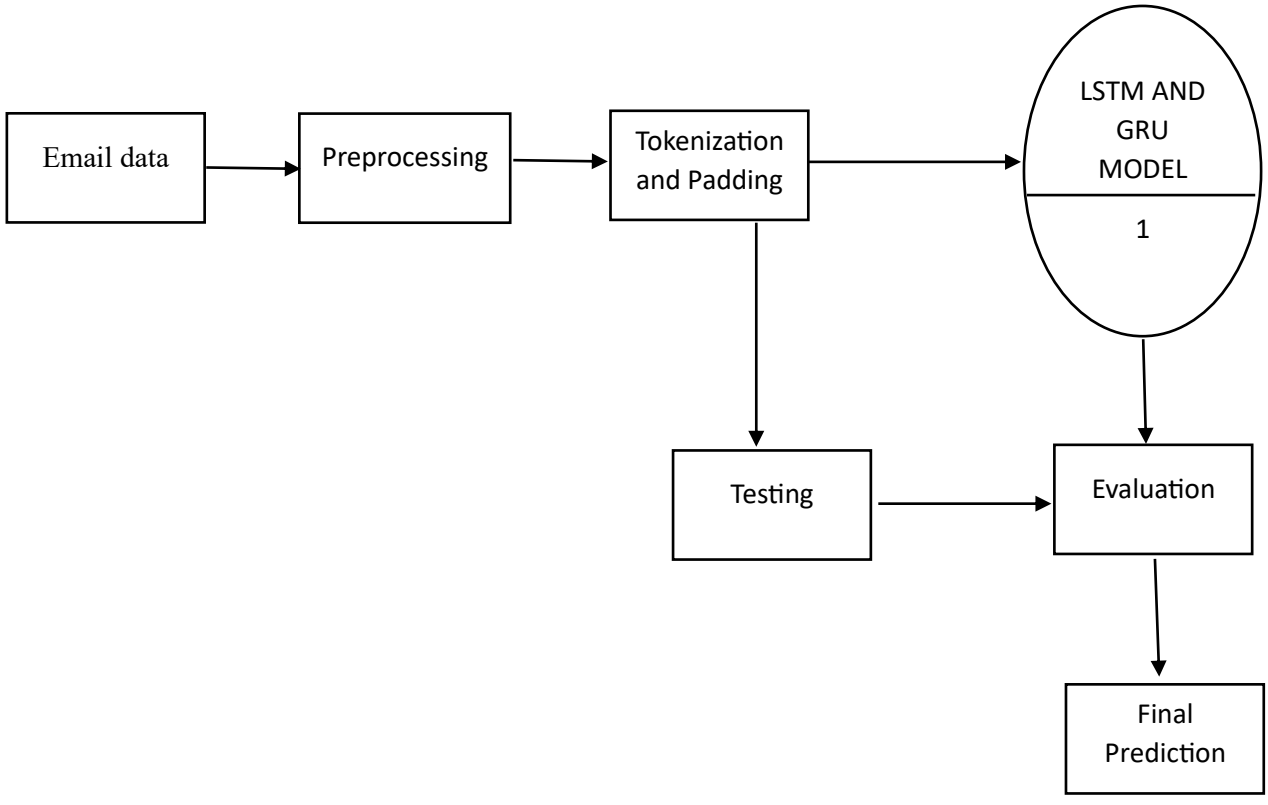


Fig 6.1.2.2.1 DFD Level 1

In the data flow diagram for email spam classification, the process begins with the reception of incoming emails in the Email Inbox. These emails undergo classification within the Email Classification System, a component designed to categorize them into four distinct classes: Normal, Fraudulent, Harassment, and Suspicious. Leveraging advanced deep learning techniques like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), this system analyzes the content and characteristics of each email to make accurate classifications. Emails categorized as Normal, indicating they are legitimate, are directed to the recipient's inbox. On the other hand, emails classified as Fraudulent, Harassment, or Suspicious, indicating potential spam or harmful content, are redirected to the Spam Folder for further review or removal. This data flow diagram illustrates the sequential flow of emails through the classification process, demonstrating how the system efficiently identifies and segregates emails based on their content to mitigate potential cyber-attacks and crimes.

6.1.3 SEQUENCE DIAGRAM

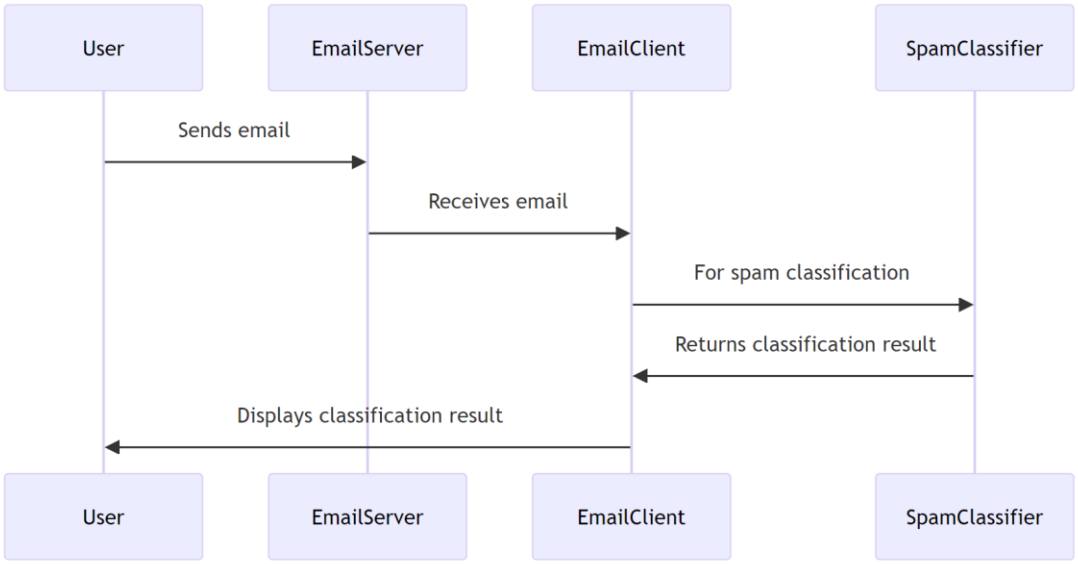


Fig 6.1.3.1 Sequence diagram

This sequence diagram encapsulates the interaction flow between the user, the email server, the email client, and the spam classifier. It begins with the user

sending an email, which is received by the email server. The email server then forwards the email to the email client. Subsequently, the email client requests the spam classifier to classify the email for spam detection purposes. The spam classifier processes the email and returns the classification result to the email client. Finally, the email client displays the classification result to the user, providing feedback on whether the email is classified as spam or not.

6.2 SYSTEM ARCHITECTURE DIAGRAM

6.2.1 TRAINING PHASE

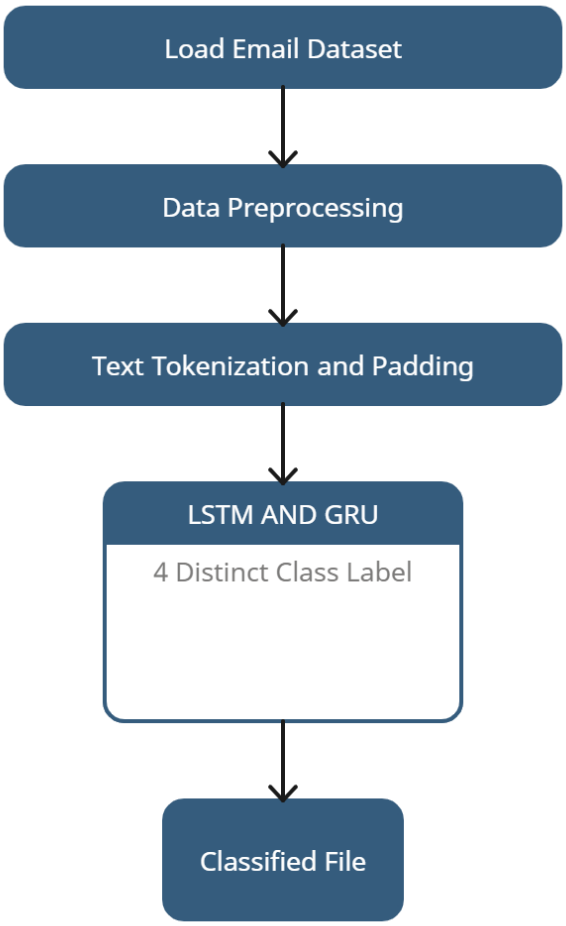


Fig 6.2.1.1 Architecture diagram for training phase

6.2.2 TESTING PHASE

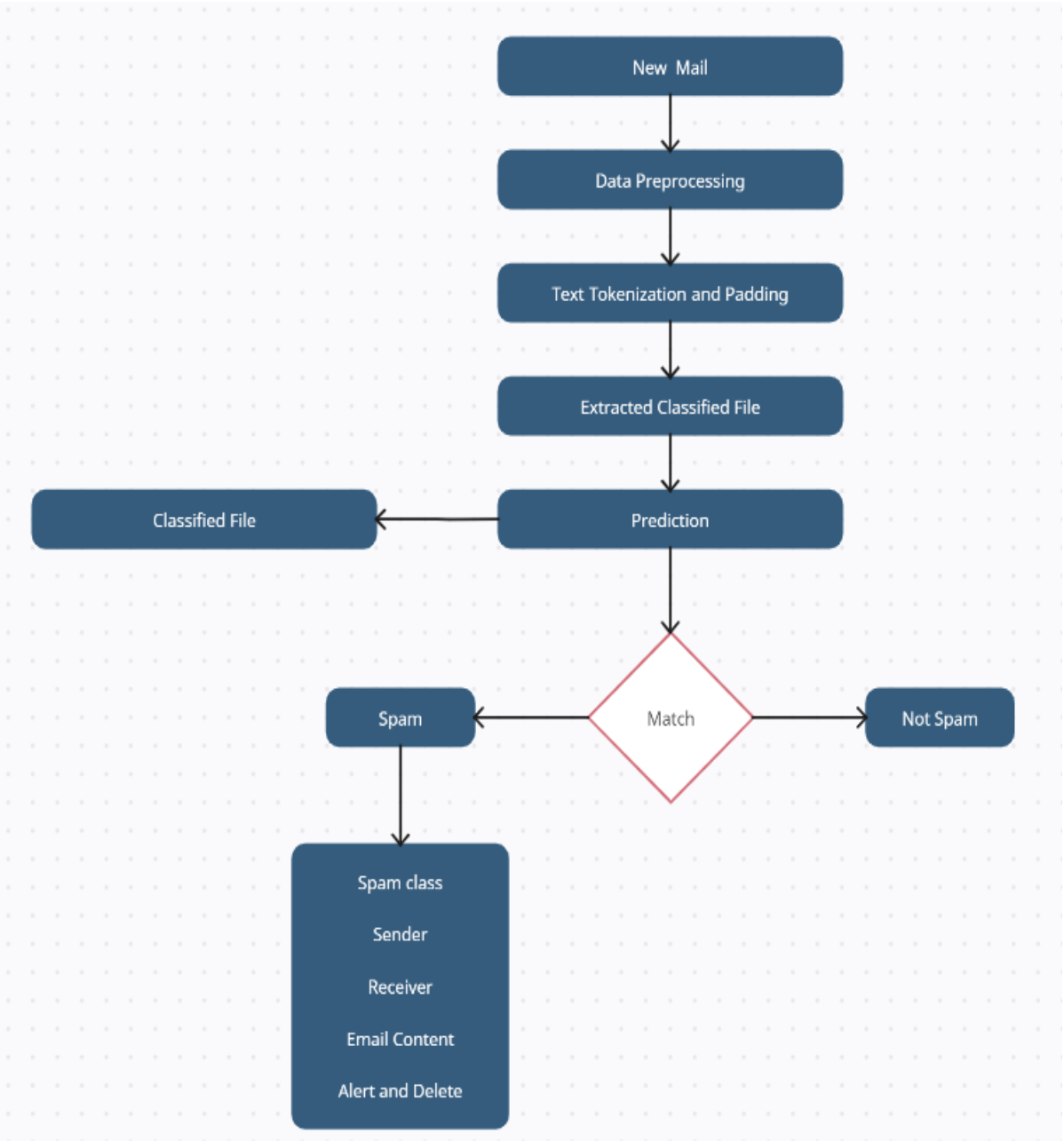


Fig 6.2.2.1 Architecture diagram for testing phase

CHAPTER 7

IMPLEMENTATION AND RESULTS

7.1 DATASET DESCRIPTION

NO.	Class Name	No. of E-Mails
1	Fraudulent	9001
2	Harassment	9138
3	Suspicious	5287
4	Normal	9001

Fig 7.1.1 Composition of Dataset

7.2 MODULES DESCRIPTION

7.2.1 DATA COLLECTION:

Email data is collected from various sources, possibly including CSV files containing email text and corresponding labels (e.g., normal, fraudulent, harassment, suspicious).

7.2.2 DATA PREPROCESSING:

Text data preprocessing is performed to clean and prepare the email text for model training.

HTML Tag Removal: BeautifulSoup library is used to remove HTML tags from the email text.

Symbol and Number Removal: Regular expressions are used to remove symbols and numbers from the text.

Lowercasing: The text is converted to lowercase to ensure uniformity.

Tokenization: The preprocessed text is tokenized using the Tokenizer class from TensorFlow's Keras API.

Padding: The tokenized sequences are padded or truncated to a fixed length to ensure uniform input size.

7.3 RESULTS

7.3.1 EVALUATION METRICS:

Loss

The loss is calculated using the categorical cross-entropy loss function, which is commonly used for multi-class classification problems. This loss function measures the dissimilarity between the true labels and the predicted probabilities assigned by the model.

The formula for categorical cross-entropy loss:

Categorical Cross-Entropy Loss = $-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ log\ (p_{ij})$

Accuracy

Accuracy = $\frac{Number\ of\ Correctly\ Classified\ Examples}{Total\ Number\ of\ Examples}$

Precision

Is the fraction of the predicted correctly classified applications to the total of all applications that are correctly real positive?

Precision=TP/TP+FP

Recall

The recall is a fraction of the predicted correctly classified applications to the total number of applications classified correctly or incorrectly.

Recall=TP/TP+FN

F-Score

F-score is the harmonic mean of precision and recall. It symbolizes the capability of the model for making fine distinctions.

F1-Score=2*(Precision*Recall)/(Precision + Recall)

7.3.2 INPUT AND OUTPUT SCREENSHOTS

7.3.2.1 INPUT FOR FRAUDULENT SPAM MAIL:

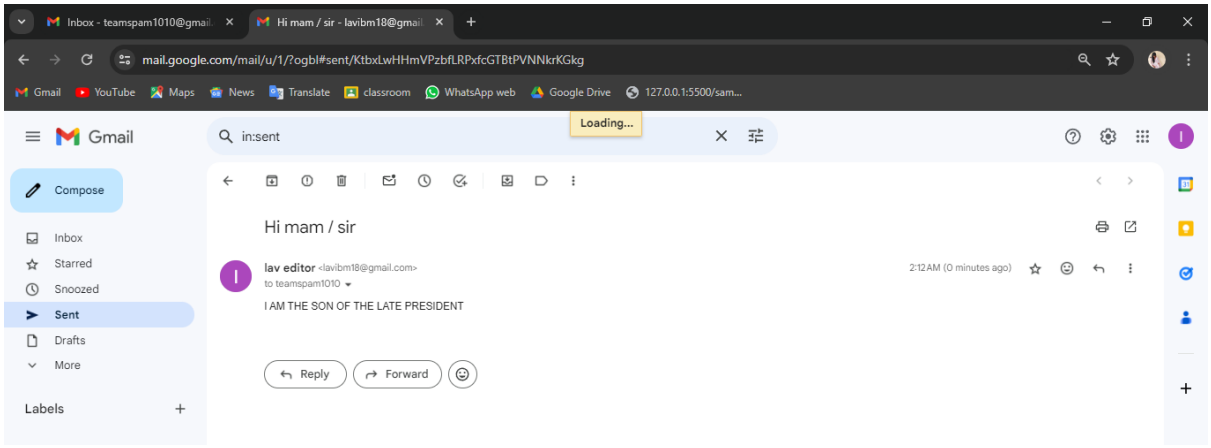
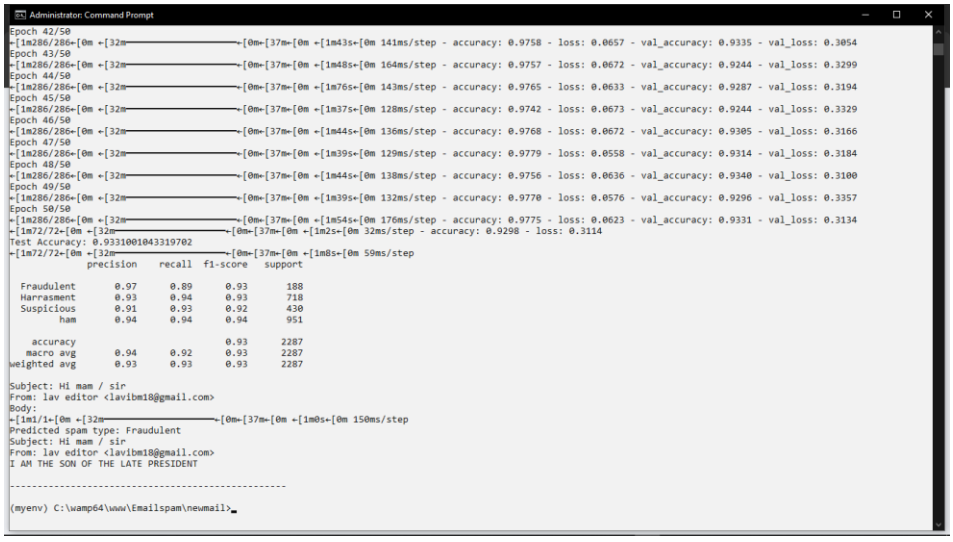


Fig 7.3.2.1.1 fraudulent spam mail

7.3.2.2 OUTPUT FOR FRAUDULENT SPAM MAIL:



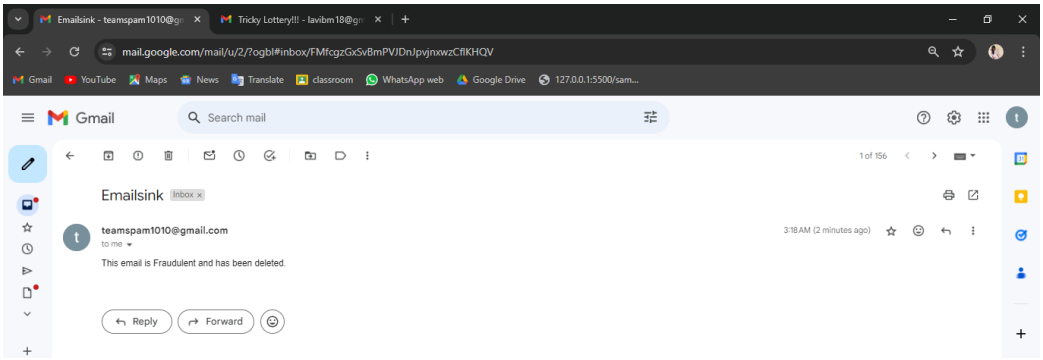


Fig 7.3.2.2.1 output for fraudulent spam mail

7.3.2.3 INPUT FOR SUSPICIOUS SPAM MAIL:

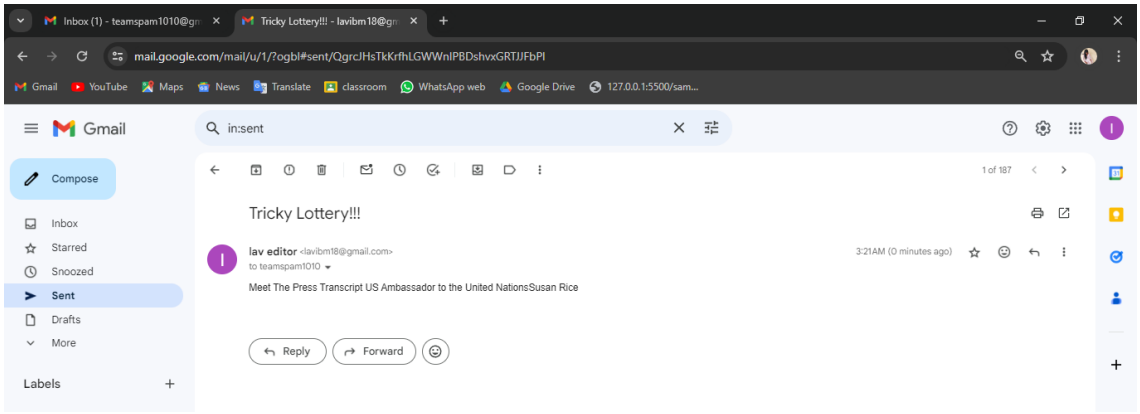
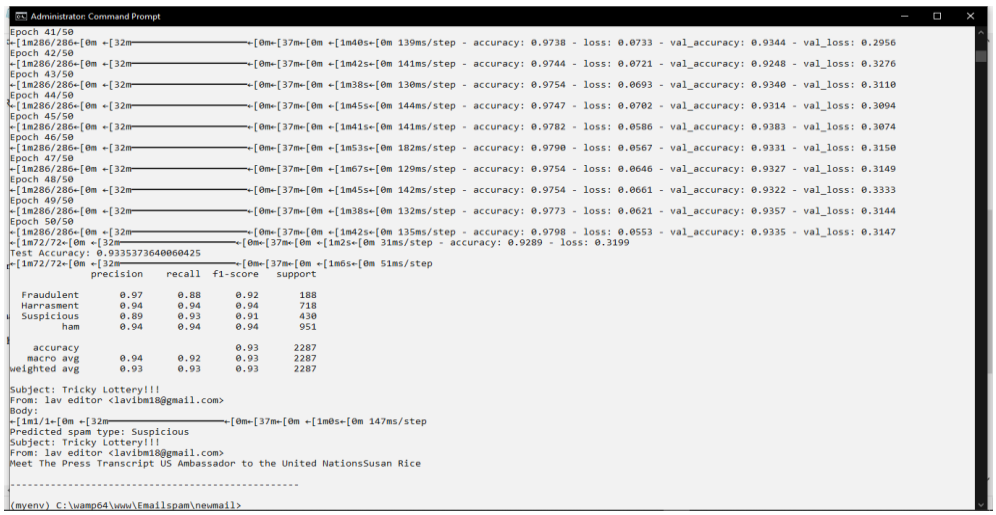


Fig 7.3.2.3.1 Suspicious spam mail

7.3.2.4 OUTPUT FOR SUSPICIOUS SPAM MAIL:



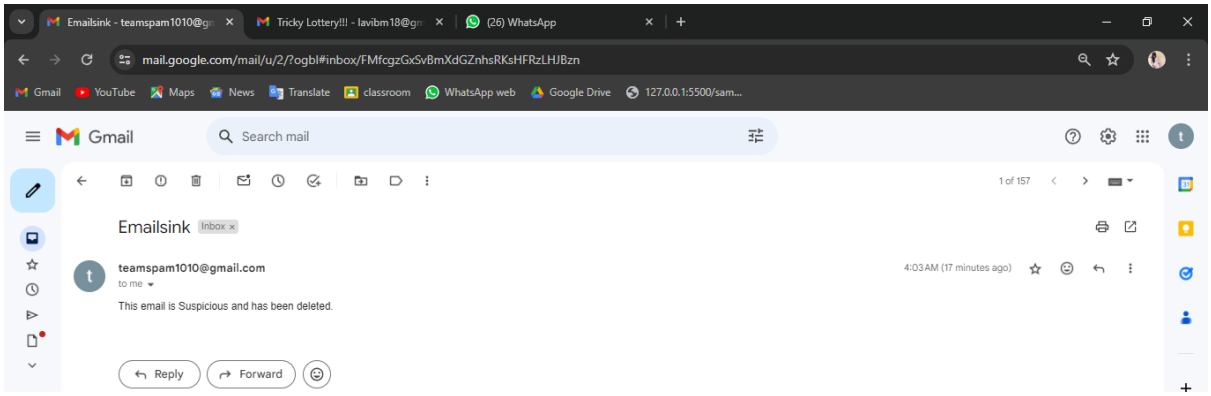


Fig 7.3.2.4.1 output for suspicious spam mail

7.3.2.5 INPUT FOR HARASSMENT SPAM MAIL:

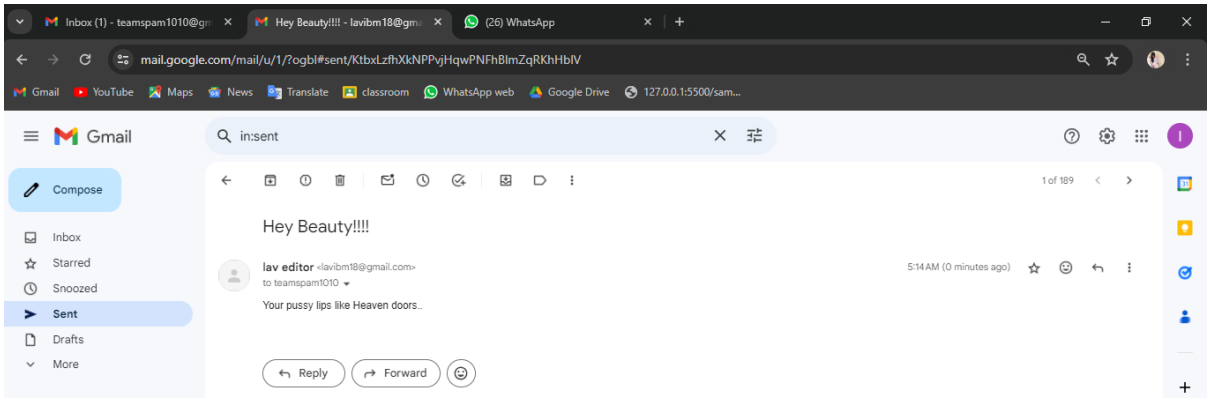
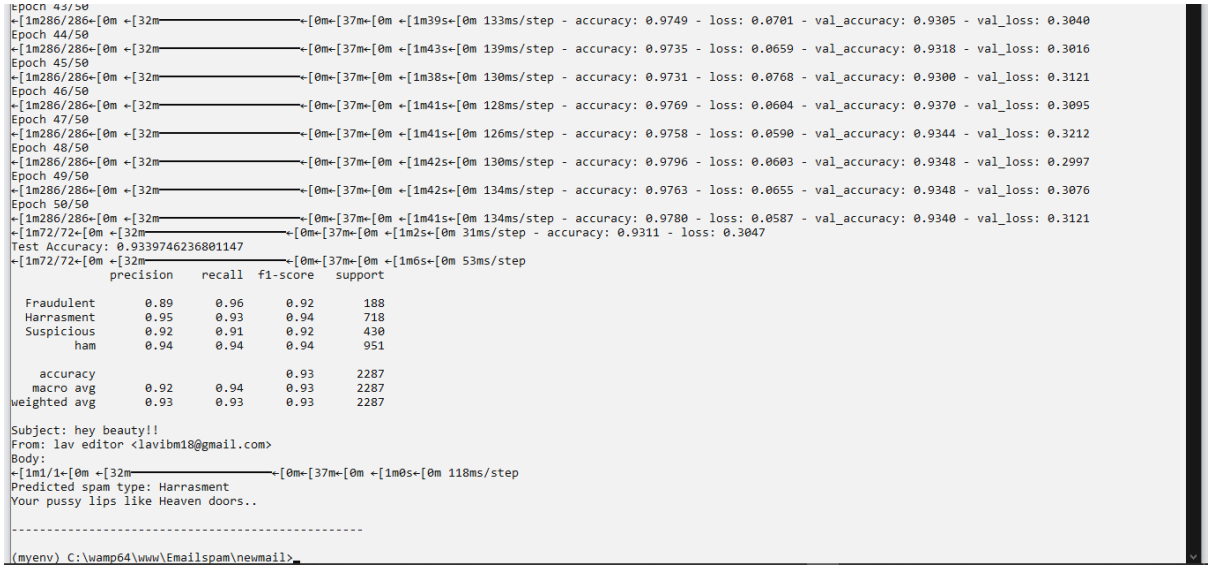


Fig 7.3.2.5.1 harassment spam mail

7.3.2.6 OUTPUT FOR HARASSMENT SPAM MAIL:



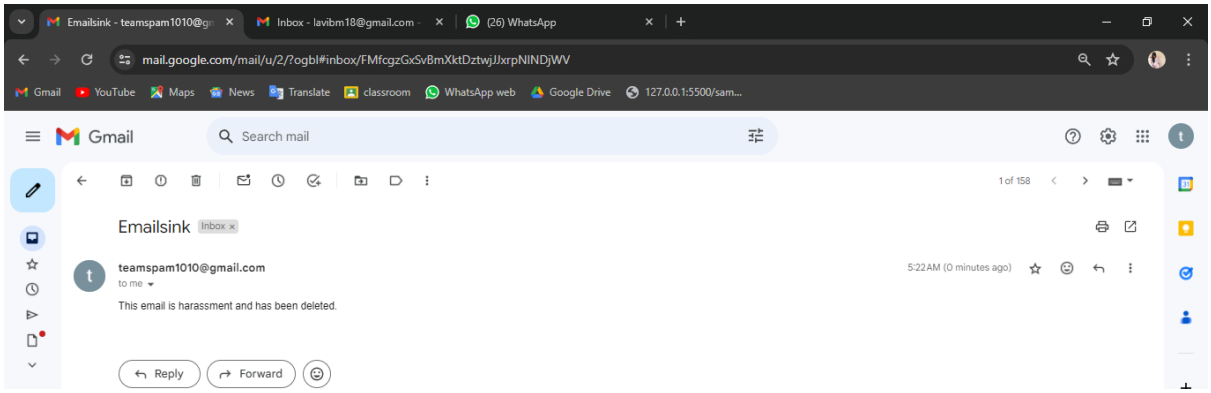


Fig 7.3.2.6.1 output for harassment spam mail

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

With the growing trend of cybercrime and accidents resulting from vulnerabilities, proactive monitoring and post-incident analysis of email data is crucial for organizations. Cybercrimes like hacking, spoofing, phishing, E-mail bombing, whaling, and spamming are being performed through E-mails. The existing email classification approaches lead towards irrelevant E-mails and/or loss of valuable information. Keeping in sight these limitations, we designed a novel efficient approach for E-mail classification into four different classes: Normal, Fraudulent, Threatening, and Suspicious E-mails by using LSTM based GRU that not only deals with short sequences as well long dependencies of 1000C characters. We evaluated the proposed model using evaluation metrics such as precision, recall, accuracy, and f-score. Experimental results revealed that Model performed better than existing ML algorithms and achieved a classification accuracy of 95% using the novel technique of LSTM with recurrent gradient units.

8.2 FUTURE ENHANCEMENT

For now, we are considering e-mail classes such as normal, harassment, fraudulent, and suspicious; however, many other classes can be added to this work in the presence of the massive amount of e-mail data.

APPENDIX - 1

SOURCE CODE

```
#EMAIL_SPAM_DETECTION

import numpy as np

import pandas as pd

import re

from bs4 import BeautifulSoup

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, GRU, Dense, Embedding, SpatialDropout1D

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad_sequences

from email.header import decode_header

from email.mime.text import MIMEText

import imaplib

import email

import smtplib

from imapclient import IMAPClient

from sklearn.metrics import classification_report

def preprocess_text(text):
```

```
    if isinstance(text, str):
        text = BeautifulSoup(text, "html.parser").get_text()
        text = re.sub(r'[^a-zA-Z\s]', '', text)
        text = text.lower()
    return text
def load_and_preprocess_data(csv_file, encoding='utf-8'):
    data = pd.read_csv(csv_file, encoding=encoding)
    data = data.dropna(subset=['text'])
    data['text'] = data['text'].apply(preprocess_text)
    return data
def delete(EMAIL, PASSWORD, IMAP_SERVER, IMAP_PORT=993):
    mail = imaplib.IMAP4_SSL(IMAP_SERVER, IMAP_PORT)
    mail.login(EMAIL, PASSWORD)
    mail.select('inbox')
    status, data = mail.search(None, 'SEEN')
    most_recent_email_id = data[0].split()[-1]
    status, message_data = mail.fetch(most_recent_email_id, '(RFC822)')
    raw_email = message_data[0][1]
    msg = email.message_from_bytes(raw_email)
    subject = decode_header(msg['Subject'])[0][0]
    sender = decode_header(msg['From'])[0][0]
    if isinstance(subject, bytes):
```

```
        subject = subject.decode()
    if isinstance(sender, bytes):
        sender = sender.decode()
    print('Subject:', subject)
    print('From:', sender)
    for part in msg.walk():
        if part.get_content_type() == "text/plain":
            content = part.get_payload(decode=True)
            charset = part.get_content_charset()
            if charset:
                content = content.decode(charset)
            else:
                content = content.decode()
    mail.store(most_recent_email_id, '+FLAGS', '\\Deleted')
    mail.expunge()
def send_confirmation_email(username, password, recipient, message):
    msg = MIMEText(message)
    msg['From'] = username
    msg['To'] = recipient
    msg['Subject'] = "Emailsink"
    with smtplib.SMTP('smtp.gmail.com', 587) as server:
        server.starttls()
```

```
        server.login(username, password)
        server.send_message(msg)
def process_unseen_email(email_text, msgid, username, password):
    preprocessed_text = preprocess_text(email_text)
    sequence = tokenizer.texts_to_sequences([preprocessed_text])
    padded_sequence = pad_sequences(sequence, maxlen=maxlen)
    class_probabilities = model.predict(padded_sequence)
    predicted_class_index = np.argmax(class_probabilities)
    predicted_class = label_encoder.inverse_transform([predicted_class_index])[0]
    print("Predicted spam type:", predicted_class)
    if predicted_class == "Harassment" or predicted_class == "Suspicious" or predicted_class == "Fraudulent" or predicted_class == "spam":
        delete(EMAIL, PASSWORD, IMAP_SERVER, IMAP_PORT=993)
        send_confirmation_email(username, password, username, f"This email is {predicted_class} and has been deleted.")
    elif predicted_class == "ham":
        return
ham_data = load_and_preprocess_data("ham.csv", encoding='latin-1')
suspicious_data = load_and_preprocess_data("suspicious.csv", encoding='latin-1')
harassment_data= load_and_preprocess_data("harassment.csv", encoding='latin-1')
```

```
fraudulent_data = load_and_preprocess_data("fraudulent.csv", encoding='latin-1')

all_data = pd.concat([suspicious_data, harassment_data, fraudulent_data, ham_data], ignore_index=True)

max_words = 1000

tokenizer = Tokenizer(num_words=max_words, filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n', lower=True)

tokenizer.fit_on_texts(all_data['text'])

sequences = tokenizer.texts_to_sequences(all_data['text'])

maxlen = 50

X = pad_sequences(sequences, maxlen=maxlen)

y = all_data['label']

label_encoder = LabelEncoder()

y = label_encoder.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

embedding_dim = 50

model = Sequential([

Embedding(input_dim=max_words,output_dim=embedding_dim,input_length=maxlen),

    SpatialDropout1D(0.2),

    LSTM(units=32,dropout=0.2,recurrent_dropout=0.2,return_sequences=True),

    GRU(units=32, dropout=0.2, recurrent_dropout=0.2, return_sequences=True),
```

```
    LSTM(units=32, dropout=0.2, recurrent_dropout=0.2),
    Dense(4, activation='softmax')  # 3 output neurons for 3 categories])
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',
metrics=['accuracy'])
epochs = 10
batch_size = 32
model.fit(X_train,y_train,epochs=epochs,batch_size=batch_size,validation_data
=(X_test, y_test))
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Accuracy:", accuracy)
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
target_names = label_encoder.classes_
print(classification_report(y_test, y_pred_classes, target_names=target_names))
IMAP_SERVER = 'imap.gmail.com'
EMAIL = 'teamspam1010@gmail.com'
PASSWORD = 'zjlezvhaysqtpumi'
mail = imaplib.IMAP4_SSL(IMAP_SERVER)
mail.login(EMAIL, PASSWORD)
mail.select('inbox')
status, data = mail.search(None, 'UNSEEN')
if status == 'OK':
```

```
    for num in data[0].split():
        status, message_data = mail.fetch(num, '(RFC822)')
        if status == 'OK':
            raw_email = message_data[0][1]
            msg = email.message_from_bytes(raw_email)
            subject = decode_header(msg['Subject'])[0][0]
            from_ = decode_header(msg['From'])[0][0]
            if isinstance(subject, bytes):
                subject = subject.decode()
            if isinstance(from_, bytes):
                from_ = from_.decode()
            print(f"Subject: {subject}")
            print(f"From: {from_}")
            for part in msg.walk():
                content_type = part.get_content_type()
                content_disposition = str(part.get('Content-Disposition'))
                if content_type=='text/plain' and 'attachment' not in
content_disposition:
                    body = part.get_payload(decode=True)
                    charset = part.get_content_charset()
                    if charset:
                        body = body.decode(charset)
```

```
                print("Body:")
                process_unseen_email(body, num, EMAIL, PASSWORD)
                print(body)
                print("-" * 50)
else:
    print("Failed to retrieve emails.")
mail.close()
mail.logout()
```

REFERENCES

[1] Asif Karim , (Member, IEEE), Sami Azam , (Member, IEEE),Bharanidharan Shanmugam , and Krishnan Kannoorpatti, 2021 -An Unsupervised Approach for Content-Based Clustering of Emails Into Spam and Ham Through Multiangular Feature Formulation.

[2] Manoj Sethi, Sumesha Chandra, Vinayak Chaudhary, Yash. 2021, Email Spam Detection using Machine Learning and Neural Networks, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072.

[3] P.U. Anitha, Dr.C.V. Guru Rao, Dr. D. Suresh Babu, 2021, Email Spam Filtering Using Machine Learning Based Xgboost Classifier Method. Turkish Journal of Computer and Mathematics Education Vol.12 No.11 (2021), 2182-2190 Research article.

[4] Zeeshan Bin Siddique,Mudassar Ali Khan,Ikram Ud Din,Ahmad Almogren,Irfan Mohiuddin and Shah Nazir. 2021, Machine Learning-Based Detection of Spam Emails. Hindawi,Scientific Programming,Volume 2021, Article ID 6508784, 11 pages https://doi.org/10.1155/2021/6508784, Research article.

[5] Ganiev Salim Karimovich, Khamidov Sherzod Jaloldin ugli, Olimov Iskandar Salimbayevich, 2020. Analysis of machine learning methods for filtering spam messages in email services. Authorized licensed use limited to: San Francisco State Univ. IEEE Xplore, DOI: 10.1109/ICISCT50599.2020.9351442.

[6] Nikhil Kumar, Sanket Sonowal, Nishant, 2020, Email Spam Detection Using Machine Learning Algorithms. Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020) IEEE Xplore Part Number: CFP20N67-ART; ISBN: 978-1-7281-5374-2.

[7] Thashina Sultana, K A Sapnaz, Fathima Sana, Mrs. Jamedar Najath.2020, Email based Spam Detection. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS060087 Published by :www.ijert.orgVol. 9 Issue 06, June-2020.

[8] Asma Bibi1, Rasia Latif, Samina Khalid, Waqas Ahmed, Raja Ahtsham Shabir, Tehmina Shahryar, 2020, Spam Mail Scanning Using Machine Learning Algorithm. Volume 15, Number 2, March 2020.

[9] Vinodhini.M, Prithvi.D, Balaji.S, 2020, Spam Detection Framework using ML Algorithm. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020.

[10] Smita sindhu, Sunil Parameshwar Patil, Arya Sreevalsan, Faiz Rahman. 2020, Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation. Authorized licensed use limited to: UNIVERSITY OF WESTERN ONTARIO, IEEE Xplore. International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE 2020).

[11] Emmanuel Gbenga Dada, Joseph Stephen Bassi,Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa, 2019. Machine learning for email spam filtering: review, approaches and open research problems. Contents lists available at ScienceDirect, Heliyon, journal homepage: www.heliyon.c.