ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023

# AI based Spam Spoiler for Public and Private E-Mail Services

[1] Savitiri.S,[2] Sowmiya.K

[1][2] Department of MCA, Dhanalakshmi Srinivasan college of engineering and Technology.

[1] mailtodilip82@gmail.com,[2] sowmiyak.mca2021@dscet.ac.in.

***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, this project proposed to design a novel efficient approach named E-mailSinkAI for E-mail classification into four different classes: Normal, Fraudulent,***

***Threatening, and Suspicious E-mails by using LSTM based GRU. The LSTM based GRU efficiently captures meaningful information from E-mails that can be used for forensic analysis as evidence. E-mailSinkAI effectively outperforms existing methods while keeping the classification process robust and reliable.***

## 1. INTRODUCTION

Overview

Email stands for Electronic Mail. It is a method to sends messages from one computer to another computer through the internet. It is mostly used in business, education, technical communication, document interactions. It allows communicating with people all over the world without bothering them. In 1971, a test email sent Ray Tomlinson to himself containing text.

**Figure 1.1. E-Mail**

Email messages are conveyed through email servers; it uses multiple protocols within the TCP/IP suite. For example, SMTP is a protocol, stands for simple mail transfer protocol and used to send messages whereas other protocols IMAP or POP are used to retrieve messages from a mail server. If you want to login to your mail account, you just need to enter a valid email address, password, and the mail servers used to send and receive messages.

Although most of the webmail servers automatically configure your mail account, therefore, you only required to enter your email address and password. However, you may need to manually configure each account if you use an email client like Microsoft Outlook or Apple Mail. In addition, to enter the email address and password, you may also need to enter incoming and outgoing mail servers and the correct port numbers for each one.

Email messages include three components, which are as follows:

- Message envelope: It depicts the email's electronic format.
- Message header: It contains email subject line and sender/recipient information.
- Message body: It comprises images, text, and other file attachments.

### 1.2. Objective of the Project

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.
To select the best model for E-mail forensic tools.
To propose a novel efficient approach named E-MailSinkAPI that uses Long Short-Term Memory (LSTM) based Gated Recurrent Neural Network (GRU) for multiclass email classification.
To detect any harmful or unfavourable e-mails received at the e-mail server end based on the deep learning-based architecture.
To model emails at the email header, the email body, the character level, and the word level simultaneously.
To distinguish whether the email is a cybercrime email.

### 1.3. About of the Project

The existing email classification approaches lead towards irrelevant E-mails and/or loss of valuable information.
The Scope of the project is to design a novel efficient approach named E-MailSinkAPI 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. E-MailSinkAPI focuses on tuning LSTM based GRU parameters to attain the best performance. The LSTM based GRU efficiently captures meaningful information from E-mails that can be used for forensic analysis as evidence. E-mail content analysis helps spoof identification since it is more efficient to analyze the headers of specific E-mails than all E-mails.

## SYSTEM ANALYSIS

### 3.1. Existing System

**1. Content-Based Filtering Technique**

Algorithms analyse words, the occurrence of words, and the distribution of words and phrases inside the content of e-mails and segregate them into spam non-spam categories.

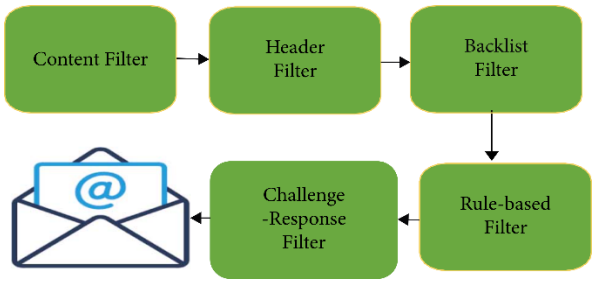



Figure 3.1. Content based Email Filtering

**2. Case Base Spam Filtering Method**

Algorithms trained on well-annotated spam/non-spam marked emails try to classify the incoming mails into two categories.

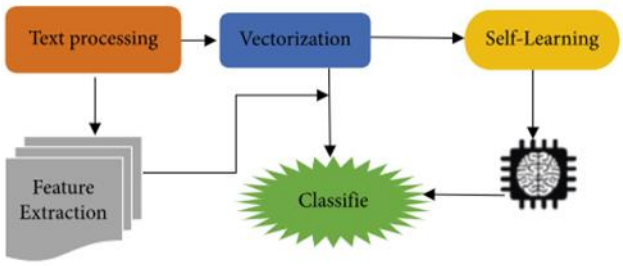



Figure 3.2. Case based Spam Filtering

### 3. Heuristic or Rule-Based Spam Filtering Technique

Algorithms use pre-defined rules in the form of a regular expression to give a score to the messages present in the e-mails. Based on the scores generated, they segregate emails into spam non-spam categories.

### 4. The Previous Likeness Based Spam Filtering Technique

Algorithms extract the incoming mails' features and create a multi-dimensional space vector and draw points for every new instance. Based on the KNN algorithm, these new points get assigned to the closest class of spam and non-spam.

### 5. Adaptive Spam Filtering Technique

Algorithms classify the incoming mails in various groups and, based on the comparison scores of every group with the defined set of groups, spam and non-spam emails got segregated.

### 6. Machine learning classifiers

The machine learning models are selected based on their group, diversity and acceptance

in the machine learning community. SVM, Naive Bayes (NB) and DT are from three different groups of classifiers.

- Support vector machine

SVM are based on the assumption that the input data can be linearly separable in a geometric space. This is often not the case when working with real word data. To solve this problem SVM map the input to a high dimension feature space, i.e hyperplane, where a linear decision boundary is constructed in such a manner that the boundary maximises the margin between two classes. SVM is introduced as a binary classifier intended to separate two classes when obtaining the optimal hyperplane and decision boundary.

- **Decision tree**

A DT classifier is modelled as a tree where rules are learned from the data in a if-else form. Each rule is a node in the tree and each leaf is a class that will be assigned to the instance that fulfil all the above nodes conditions. For each leaf a decision chain can be created that often is easy to interpret. The interpretability is one of the strengths of the DT since it increases the understanding of why the classifier made a decision, which can be difficult to achieve with other classifiers. The telecommunication company is today using a manually created decision tree, in which the rules are based on different combinations of words.

- **Naive bayes**

The NB classifier is considered to perform optimal when the features are independent of ach other, and close to optimal when thefeatures are slightly dependant. Real world ata does often not meet this criterion but esearchers have shown that NB perform better or similar to C4.5, a decision tree algorithm in some settings. The researchers argue that NB performs well even when there is a clear dependency between the features, making it applicable in a wide range of tasks.

- **AdaBoost**

ADA is built upon the premise that multiple weak learners that perform somewhat good can be combined using boosting to achieve better result. This algorithm performs two important steps when training and combining the weak classifiers, first it decided which training instances each weak classifier should be trained on, and then it decides the weight in the vote each classifier should have. Each weak classifier is given a subset of the training data that where each instance in the training data is given a probability that is decided by the previous

ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023

weak classifiers performance on that instance. If the previous weak classifiers have failed to classify the instance correct it will have a higher probability to be included in the following training data set. The weight used in the voting is decided by each classifiers ability to correctly classify instances. A weak classifier that performs well is given more influence than a classifier that perform bad.

**Disadvantages**

- The existing email classification approaches lead towards irrelevant E-mails and/or loss of valuable information.
- The blocklisting process focuses on identifying and documenting individuals, which takes a lot of manpower and time.
- ML algorithms also need manual feature engineering for the representation of features that are not very conducive.
- Forensic tools often lead to irrelevant E-mails as they use keyword search-based methods.

### 3.2. Proposed System

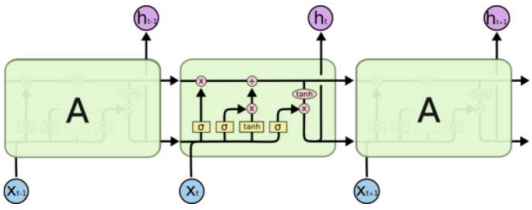
The proposed approach comprises data collection, pre-processing, feature extraction, parameter tuning, and classification using the LSTM-GRU model. In this project, E-mail datasets are divided into normal, harassing, suspicious, and fraudulent classes. The E-mail is divided into word levels of the E-mail body, and the embedding layer is applied to train and obtain the sequence of vectors.

- **LSTM and GRU**

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

- **LSTM – Long Short-Term Memory**

LSTMS 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 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.

These cells use the gates to regulate the information to be kept or discarded at loop operation before passing on the long term and short-term information to the next cell. We can imagine these gates as Filters that remove unwanted selected and irrelevant information. There are a total of three gates that LSTM uses as Input Gate, Forget Gate, and Output Gate.

**Input Gate**

The input gate decides what information will be stored in long term memory. It only works with the information from the current input and short-term memory from the previous step. At this gate, it filters out the information from variables that are not useful.

Forget Gate

The forget decides which information from long term memory be kept or discarded and this is done by multiplying the incoming long-term memory by a forget vector generated by the current input and incoming short memory.

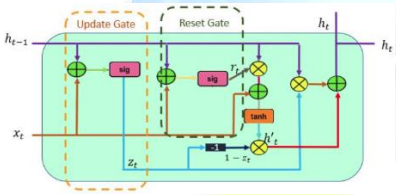
Output Gate

The output gate will take the current input, the previous short-term memory and newly computed long-term memory to produce new short-term memory which will be passed on to the cell in the next time step. The output of the current time step can also be drawn from this hidden state.

ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023

- 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.

Update gate

The update gate is responsible for determining the amount of previous information that needs to pass along the next state. This is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient.

Reset gate

The reset gate is used from the model to decide how much of the past information is needed to neglect; in short, it decides whether the previous cell state is important or not.

First, the reset gate comes into action it stores relevant information from the past time step into new memory content. Then it multiplies the input vector and hidden state with their weights. Next, it calculates element-wise multiplication between the reset gate and previously hidden state multiple. After summing up the above steps the non-linear activation function is applied and the next sequence is generated.

Advantages

- Effectively classify E-mail content.
- Keeping the classification process robust and reliable.
- E-mail content analysis helps spoof identification.
- No need of manpower for labelling.
- Detect any harmful or unfavorable e-mails received at the e-mail server end.

## PROJECT DESCRIPTION

### 6.1 Problem Statement

Many people rely on the Internet for many of their professional, social and personal activities. But there are also people who attempt to damage our Internet-connected computers, violate our privacy and render inoperable Internet services.

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.

Figure 1.2. E-Mail Attacks

Below are some of the most common types of Attacks:

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.

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.

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.

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.

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.

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.

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.

**Modules Description**

E-MailSinkAI Web App

Build an E-mail Forensics Predictor service is an online platform which people use emails free from spam. In this module we developed the Web based GUI is developed for the email classification system to categorize the email as spam or not spam. Integrate with Trainer and Tester Modules Developed with Python and Flask Framework

User Account Management

- Admin
- Email user

E-Mail Classification – Training Phase

Dataset Annotation

In this project, E-mails are divided into normal, harassing, suspicious, and fraudulent classes. The E-mail is divided into word levels of the E-mail body, and the embedding layer is applied to train and obtain the sequence of vectors.

E-mail Data Set Preparation and Exploration

- Import Dataset

In this module the admin uploads an email dataset (CSV) file. This will be used to train your email forensics analysis model.

- Read Dataset

The EmailSinkAI reads email dataset to output the purpose or objective of the project.

- Explore Dataset – EDA

Data visualization tool that brings the entirety of data together into a striking and easy-to-follow view.

Data Pre-processing

The data pre-processing phase consists of natural language-based steps that standardize the text and prepare it for analysis.

- Tokenization

Breaking up the original text into component pieces is the tokenization step in natural language processing. There are predefined rules for tokenization of the documents into words. The tokenization step is performed in Python by using the SpaCy library.

- Normalization

These are the steps needed for translating text from human language (English) to machine-readable format for further processing. The process starts with:

- changing all alphabets to lower or upper case
- expanding abbreviations
- excluding numbers or changing those to words
- removing white spaces
- removing punctuation, inflection marks, and other circumflexes
- removing stop words, sparse terms, and particular words
- text canonicalization

- Stop Words Removal

Words like ``a'' and ``the'' that appear so frequently are not relevant to the context of the E-mail and create noise in the text data. These words are called stop words, and they can be filtered from the text to be processed. We utilized the``NLTK'' Python library to remove stop words from the text.

- Punctuation Removal

Punctuation includes (e.g., full stop (.), comma (,), brackets) to separate sentences and clarify meaning. For punctuation removal, we utilize the ``NLTK'' library.

Feature Extraction

After eliminating irrelevant information, the elaborated list of words is converted into numbers. The TF-IDF method is applied to accomplish this task. Term Frequency is several occurrences of a word in a document, and IDF is the ratio of a total number of documents and the number of documents containing the term. A popular and straightforward method of feature extraction with text data is called the bag-of-words model of text. A bag-of-words model, or BoW for short, is a way of extracting features from the text for use in modelling, such as machine learning algorithms. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things (1) A vocabulary of known words, (2) A measure of the presence of known words. We extract features on the basis of Equations Here tf represents term frequency and df represents document frequency.

$$TFIDF = tf * (\frac{1}{df})$$
$$TFIDF = tf * Inverse(df)$$
$$TFIDF(t, d, D) = TF(t, d).IDF(t, D)$$
$$TFIDF(t, d) = log\frac{N}{|d \epsilon Dt \epsilon D|}$$

Eq. 1-5

Feature extraction in DL with the context of words is also essential. The technique used for this purpose is word2vec neural network-based algorithm. Equation 5 given below shows how word2vec manages the word-context with the help of probability measures. The D represents the pair-wise illustration of a set of words, and (w; c) is the word-context pair drawn from the large set D.

$$P(D = 1 \mid w, c_{1:k}) = \frac{1}{1 + e^{-(w \cdot c_1 + w \cdot c_2 + \ldots + w \cdot c_k)}}$$

Eq.5

The multi-word context is also a variant of word2vec, as shown in Equation 6. The variable-length context is also controlled by the given below mathematics

$$P(D = 1 \mid w, c) = \frac{1}{1 + e^{-s(w,c)}}$$

**Word Embedding Layer**

Embedding is the representation of words into real numbers. Many machine learning and DL Algorithms cannot process data in raw form (text form) and can only process numerical values as input for learning. Word embedding organizes texts which are converted into numbers. It extracts relevant features from the textual data and structures them up in the form of real values. Word embedding uses a word mapping dictionary to convert the terms (words) to a real value vector. There are two main problems with machine learning feature engineering techniques, one problem is the sparse vectors for data representation, and the second issue is that; it does not take into account the meaning of words to some extent. In embedding vectors, similar words will be represented by the almost near real-valued numbers. For example, the terms love and affection will be near to each other in the embedding vector.

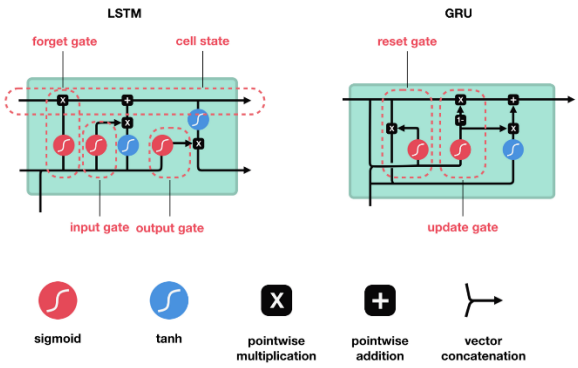
The embedding vector as a data structure in the DL algorithm is used to accomplish the learning. In the experimental setup, the word embedding layer contains information about the sequence length of E-mails. We consider the sequence length of the E-mail 600 characters each. The embedding dimensions used in SeFACED is 800. The vocabulary size is set to 70; 000 at the start because we set this value after generating the unique tokens of our dataset. The embedding layer takes three arguments such as input dimensions, output dimensions, and input length. In our proposed study, the input dimensions are 800, vocabulary size is 70; 000, and input length is 600. We need to be curious when setting the embedding layer dimensions because sometimes we skip the essential features when dealing with the large size of textual input. The embedding layer output will be used for the LSTM and GRU layers in adjacent layers.

LSTM based GRU Classification Model

LSTM comprises the classification of E-mails as Normal, Harassing, Suspicious, and Fraudulent. The LSTM and GRU are both based on the gated network architecture, due to which we combined the GRU and LSTM to utilize the gated architecture of both of them.

The DL models' layered structure helps in learning without intervention in ML model implementation. Several libraries provide an in-depth learning implementation structure. We split the data into three training, validation, and testing sets with a 65 V 10 V 25 ratio. We extracted the features from textual data of E-mail using the word Embedding technique. We encode the target values using the one-hot encoding technique into 4-distinct classes. We pass all pre-processed data to the novel architecture of LSTM layers variants for the perfect classification of E-mails. We use the LSTM layers with different GRU and Convo1D layer variants to transform the input textual data into an efficient E-mail classification system.

ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023




Textual data needs special attention when feature extraction comes in the proposed methodology. Different feature extraction methods need to be implemented when solving the Natural language processing problem using DL. The main point is to convert the textual data into real-valued vectors. There is a unique name for the vector in natural language processing, ``embedding vector''. There are multiple ways to generate the embedding vector from the textual data, but famous methods are GLOVE and Word2Vec techniques. Embedding vector dimensions are essential to get all the features extracted from the data. Let us suppose if we have 8 samples of textual data. The data have two distinct classes. Each sample has a maximum of five tokens in it. The vocabulary size will be the unique words in 8 samples, and the vocabulary size needs to be higher than the available unique tokens in the dataset to avoid collisions with a hash function. In this case, the dimensions of the embedding vector will be 4 x 8. In the case of the classification problem of NLP, we need to encode the target values using the one-hot encoding method. After getting the vectors from the words, the similarity between the words is measured using the similarity measure between the corresponding vectors using Equation 7.

$$\text{sim}_{\cos}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} = \frac{\sum_i u_{[i]} \cdot v_{[i]}}{\sqrt{\sum_i (u_{[i]})^2} \sqrt{\sum_i (v_{[i]})^2}}$$

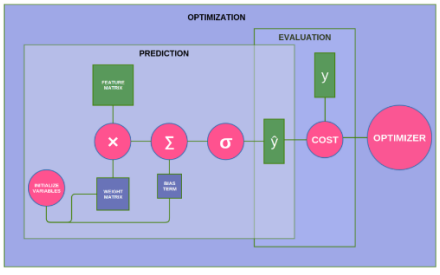
Eq.7

There are many other ways to measure the similarity between the word vectors, one of them is Jaccard similarity, which defines Equation 8.

$$\text{sim}_{\text{Jacard}}(u, v) = \frac{\sum_i \min(u_{[i]}, v_{[i]})}{\sum_i \max(u_{[i]}, v_{[i]})}$$

DL for NLP uses dense vector representation to reduce the memory requirement for large models. Dense vector categorical data encoding is also a famous method, but most literature is based on one-hot encoding techniques. After feature extraction, the language modelling phase comes up.

**E-Mail Forensic Analyser – Testing Phase**



**Email Account Integrator**
In this module, user register here and login with their username and password and input their email ID and email Password to filter spam.
**Read Mail**
This module reads the new mails
**Extract Feature**
Extract spam feature from the new mails
**E-Mail Forensic Predictor**
This module predicts whether it is spam or not spam and classify the type.

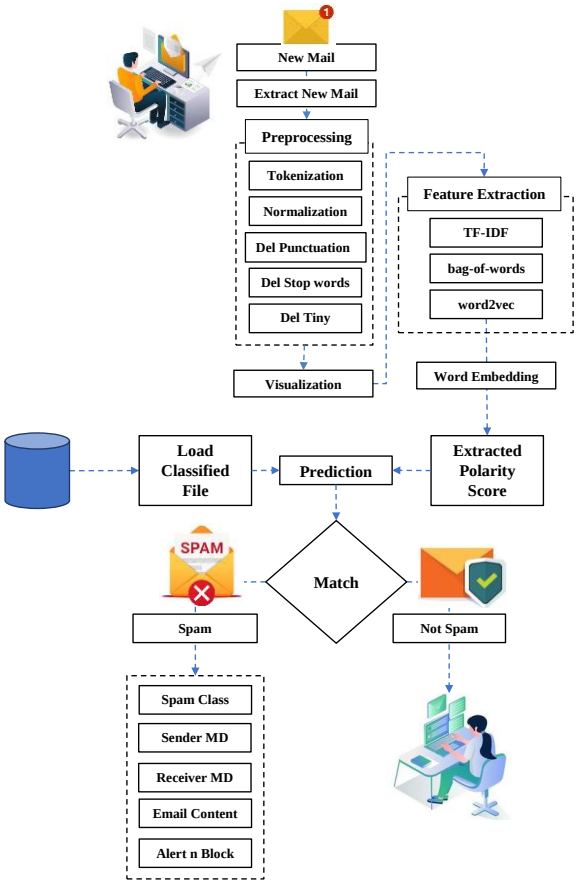

**Decision Making**
Block the mail and send alert message to the user if the mail is spam.

**6.4 Input Design**
The input design for the Spam SpoilerAPI involves defining the format and structure of the input data that the system will receive. Here are some considerations for the input design:
1. Data Format: The input data for the Spam SpoilerAPI would be email messages that need to be classified as Normal, Fraudulent, Harassment, or Suspicious. The email messages can be in CSV formats. The API should be able to handle emails in different formats and extract the relevant text for classification.
2. Data Source: The input data for the Spam SpoilerAPI can come from various sources, including private and public email services, web forms, or other communication channels. The system should be able to receive input data from different sources and classify them accurately.
3. Data Validation: The input data should be validated to ensure that it is in the correct format and does not contain any malicious content that can harm the system. The API should check for common email vulnerabilities, such as buffer overflows, SQL injections, or cross-site scripting attacks.
4. API Interface: The input data should be transmitted to the API via a secure interface, such as HTTPS, to prevent unauthorized access or interception. The API interface should also support authentication and authorization mechanisms to control access to the system.
5. Input Processing: Once the input data is received, the Spam SpoilerAPI should preprocess the data by removing any unnecessary information like email addresses, URLs, and special characters. The text should be converted to lowercase, and stop words should be removed before the classification process.
6. Response Generation: After the input data is processed and classified, the Spam SpoilerAPI should generate a response indicating whether the email is Normal, Fraudulent, Harassment, or Suspicious. If the email is classified as spam, the API should delete the email and send an auto-reply to the user, informing them that the email was detected as spam and deleted to protect their inbox.

**6.5. Output Design**

The output design for the Spam SpoilerAPI involves defining the format and structure of the output data that the system will generate. Here are some considerations for the output design:
1. Data Format: The output data for the Spam SpoilerAPI would be the classification result of the input email message, indicating whether it is Normal, Fraudulent, Harassment, or Suspicious. The classification result can be in a structured format, such as JSON or XML, or a simple text format.

2. Response Codes: The API should use response codes to indicate the success or failure of the classification process. The response codes should be standardized and documented to allow
3. easy integration with other applications.
4. Error Handling: The API should provide meaningful error messages and error codes in case of classification failure or input data




bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system. In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program validation errors. The error messages should be human-readable and
5. provide suggestions on how to resolve the issue.
6. Auto-Reply: If the input email message is classified as spam and deleted, the API should generate an auto-reply message to the sender, indicating that the email was detected as spam and deleted to protect the user's inbox. The auto-reply message

## CHAPTER 7
## SYSTEM TESTING

### TESTING DESCRIPTION
System testing is a critical aspect of Software Quality Assurance and represents he ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has to be checked separately.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

Peak Load Tests: This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.

Storage Testing's This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.

Performance Time Testing's This test determines the length of the time used by the system to process transaction data.

In this phase the software developed Testing is exercising the software to uncover errors and ensure the system meets defined requirements. Testing may be done at 4 levels

- Unit Level
- Module Level
- Integration & System
- Regression

**UNIT TESTING**

A Unit corresponds to a screen /form in the package. Unit testing focuses on verification of the corresponding class or Screen. This testing includes testing of control paths, interfaces, local data structures, logical

decisions, boundary conditions, and error handling. Unit testing may use Test Drivers, which are control programs to co-ordinate test case inputs and outputs, and Test stubs, which replace low-level modules. A stub is a dummy subprogram.

**MODULE LEVEL TESTING**

Module Testing is done using the test cases prepared earlier. Module is defined during the time of design.

**INTEGRATION & SYSTEM TESTING**

Integration testing is used to verify the combining of the software modules. Integration testing addresses the issues associated with the dual problems of verification and program construction. System testing is used to verify, whether the developed system meets the requirements.

**REGRESSION TESTING**

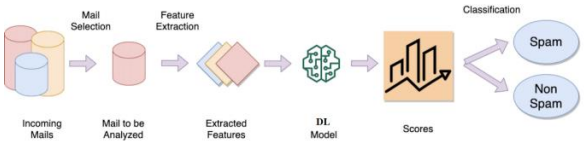
Each modification in software impacts unmodified areas, which results serious injuries to that software. So the process of re-testing for rectification of errors due to modification is known as regression testing. Installation and Delivery Installation and Delivery is the process of delivering the developed and tested software to the customer. Refer the support procedures Acceptance and Project Closure Acceptance is the part of the project by which the customer accepts the product. This will be done as per the Project Closure, once the customer accepts the product; closure of the project is started. This includes metrics collection, PCD, etc….

**SYSTEM IMPLEMENTATION**

**Problem Description**

An email message primarily contains two main components: the header and the body. The header part contains broader content-related information such as the subject, sender, recipient email addresses, and timestamp. The heart of the email is its body comprising variable contents that vary from message to message. For example, it may include a web page, files, analog data, images, audio, video, and HTML markup, among other innards. The header line usually commences with a "From" field. The email undergoes some alteration as it moves from one server to another through intermediate servers. Users may inspect headers to view the email adhered's routing path. The contents may go through pre-processing before they serve as an input to the classifier. We extract email messages for classification purposes. Uniform Resource Locator (URLs), Hypertext Markup Language (HTML), and cascading style sheets (CSS), JavaScript code, special symbols are removed first from the email message, leaving unformatted text only in the stage called pre-processing**.**

ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023




We first proceed to the standard ML algorithm. For text-based classification, the NLTK library tokenized the message text into terms where each term got quantized using the TFIDF technique. Within the proposed framework, the model extricates the features using TFIDF Vectorise followed by Gaussian Naïve Bayes (comes with the sklearn python library) for prediction. The case of deep machine learning is a special one. In this part, the tokenization layer maintains a dictionary that maps a word to an index. The embedding layer internally maintains a lookup table, which maps the index/token to a vector. It is the vector that represents the word in the complex dimensional space. The process appears as email →Token→lookuptable→vector. We use components such as Embedding and LSTM Layers and GRU in building our network. The purpose of using bidirectional is that the situational information often comes at the end of the sentence sometimes. Without using this information, uncertainty might ascend. The first LSTM network feeds in the input sequence as per usual. The second LSTM network reverses the input sequence and feeds that into the LSTM network. After that, the merger of these two networks served as input to the next layer. Of the two embedding options, including 1) train from scratch the Embedding Layer and 2) Employee some weight embedding pre-trained open source, we preferred the Glove word embedding from Stanford NLP Group. To apply the WordCloud embeddings, we first converted email message text to sequences. NLP then defined a vocabulary where each word had an inimitable index. We padded shorter sentences to the max length (most extended length rendered after pre-processing). After that, LSTM served to get the context representations. We further extended the testing phase by processing the emails by including and excluding digits/figures from their main body, respectively.

**Dataset Description**

The dataset used in this project is an amalgamation of four different datasets. The dataset contains Normal e-mails from Enron Corpora, Fraudulent e-mails provided by Phished e-mails corpora which contain misleading information, Harassment messages selected from Hate Speech, Offensive dataset. We enhance the dataset of Email Forensics by adding the suspicious emails data from email sources, and twitter source. The suspicious dataset contains some terrorism-related messages collected from Twitter by API. These different datasets are merged into a structural file to make the multiclass E-mail classification possible.

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

**Table 5.2.1. Composition of Dataset**

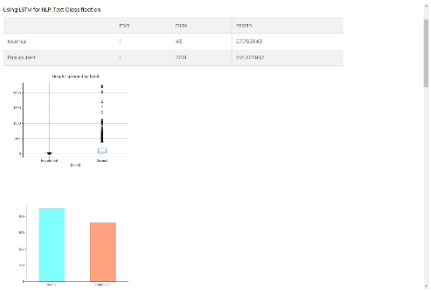
**CONCLUSION**

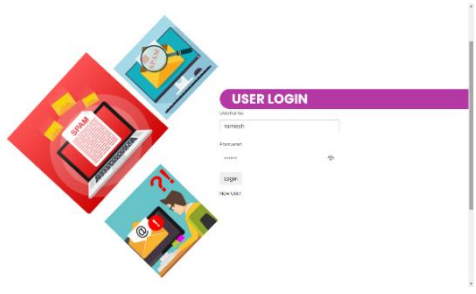
With the growing trend of cybercrime and accidents resulting from vulnerabilities, proactive monitoring and post-incident analysis of email data is crucial for organizations. Cybercrimes like hacking, spoofing, phishing, E-mail bombing, whaling, and spamming are being performed through E-mails. The existing 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 named E-MailSinkAI 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 E-MailSinkAI model using evaluation metrics such as precision, recall, accuracy, and f-score. Experimental results revealed that E-MailSinkAI performed better than existing ML algorithms and achieved a classification accuracy of 95% using the novel technique of LSTM with recurrent gradient units.

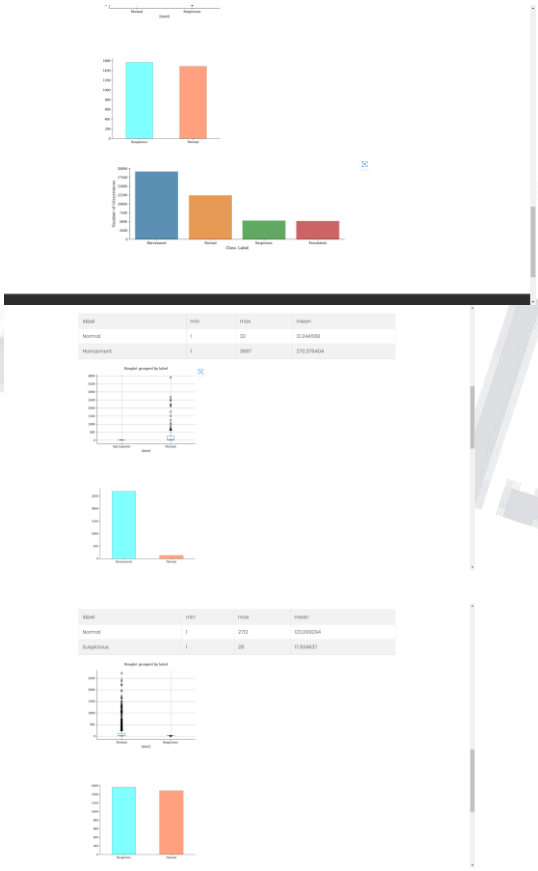
**FUTURE ENHANCEMENT**

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.

ISSN (Online): 2456-0448
International Journal of Innovative Research in Management, Engineering, and Technology
Vol. 8, Issue 6, June 2023







**REFERENCES**

1. S. Sinha, I. Ghosh, and S. C. Satapathy, ``A study for ANN model for spam classification," in Intelligent Data Engineering and Analytics. Singapore: Springer, 2021, pp. 331-343.
2. Q. Li, M. Cheng, J. Wang, and B. Sun, ``LSTM based phishing detection for big email data," IEEE Trans. Big Data, early access, Mar. 12, 2020, doi: v10.1109/TBDATA.2020.2978915.
3. T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, ``Applicability of machine learning in spam and phishing email filtering: Review and approaches," Artif. Intell. Rev., vol. 53, no. 7, pp. 5019-5081, Oct. 2020, doi: 10.1007/s10462-020-09814-9.
4. E. Bauer. 15 Outrageous Email Spam Statistics That Still Ring True in 2018, RSS. Accessed: Oct. 10, 2020. [Online]. Available: https://www.propellercrm.com/blog/email-spam-statistics.
5. A.Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, ``A comprehensive survey for intelligent spam email detection," IEEE Access, vol. 7, pp. 168261-168295, 2019.
6. K. Singh, S. Bhushan, and S. Vij, ``Filtering spam messages and mails using fuzzy C means algorithm," in Proc. 4th Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU), Apr. 2019, pp. 1-5.
7. R. S. H. Ali and N. E. Gayar, ``Sentiment analysis using unlabeled email data," in Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE), Dec. 2019, pp. 328-333.
8. K. Agarwal and T. Kumar, ``Email spam detection using integrated approach of naïve Bayes and particle swarm optimization," in Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS), Jun. 2018, pp. 685-690.
9. M. Shuaib, O. Osho, I. Ismaila, and J. K. Alhassan, ``Comparative analysis of classification algorithms for email spam detection," Int. J. Comput. Netw. Inf. Secur., vol. 10, no. 1, pp. 60-67, Aug. 2018.
10. G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, ``Email classification research trends: Review and open issues," IEEE Access, vol. 5, pp. 9044-9064, 2017.

11. Z. Chen, Y. Yang, L. Chen, L.Wen, J.Wang, G. Yang, and M. Guo, ``Email visualization correlation analysis forensics research," in Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud), Jun. 2017, pp. 339-343.
12. N. Moradpoor, B. Clavie, and B. Buchanan, ``Employing machine learning techniques for detection and classification of phishing emails," in Proc. Comput. Conf., Jul. 2017, pp. 149-156.
13. A.S. Aski and N. K. Sourati, ``Proposed efficient algorithm to filter spam using machine learning techniques," Pacific Sci. Rev. A, Natural Sci. Eng., vol. 18, no. 2, pp. 145-149, Jul. 2016.
14. Y. Kaya and Ö. F. Ertu§rul, ``A novel approach for spam email detection based on shifted binary patterns," Secur. Commun. Netw., vol. 9, no. 10, pp. 1216-1225, Jul. 2016.
15. I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, ``A combined negative selection algorithm-particle swarm optimization for an email spam detection system," Eng. Appl. Artif. Intell., vol. 39, pp. 33-44, Mar. 2015.