DEVELOPMENT OF TEXT-IMAGE SPAM DETECTION TECHNIQUES USING MULTI-MODAL RECURRENT NEURAL NETWORK AND CONVOLUTION NEURAL NETWORK TECHNIQUES

BY

ABDULLAHI, Muhammad

MTech/SICT/2018/8773

DEPARTMENT OF COMPUTER SCIENCE

FEDERAL UNIVERSITY OF TECHNOLOGY

MINNA

JANUARY, 2022

DEVELOPMENT OF TEXT-IMAGE SPAM DETECTION TECHNIQUES USING MULTI-MODAL RECURRENT NEURAL NETWORK AND CONVOLUTION NEURAL NETWORK TECHNIQUES

BY

ABDULLAHI, Muhammad

MTech/SICT/2018/8773

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL,

FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF TECHNOLOGY

(MTech) IN COMPUTER SCIENCE

JANUARY, 2022

ABSTRACT

Spam emails are unsolicited message content shared through emails to several recipients using electronic devices. Despite the emergence of alternative forms of online communication which include social networking, sending and receiving emails has remained the most convenient and time efficient method of online communication. The increase in online transactions via email has led to a significant increase in the global number of email spam which has relatively become a critical problem in the area of computing. There have been numerous techniques of machine learning for identifying unsolicited email spam. Despite the significant improvements made in the number of existing literatures, there is no classification technique that has achieve 100% accuracy, each algorithm employs a limited number of features. Thus, determining the most appropriate technique is a critical task because their effectiveness needs to be weighed relative to their drawbacks. As a result, two deep learning techniques were explored and analyzed to identify both textual and image based email spam in this study. The study is aimed to analyze the effectiveness of Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) and develop a multi-modal architecture capable of detecting textual spam, image spam and mixed spam. Enron and Image Spam Hunter email datasets were used on the test size of 30% to obtain the performance of the model. The model was trained on text-image data and achieved an accuracy of 98% detection rate which indicates that the resultant model has outperforms the other models as compared to 85% achieved by Naïve Bayes, 95% achieved by Char-CNN and 97% achieved by Support Vector Machine (SVM) respectively.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI: Artificial Intelligence

CM: Confusion Matrix

DT: Decision Tree

FP: False Negative

Ham: Legitimate Emails

ML: Machine Learning

CNN: Convolution Neural Network

RNN: Recurrent Neural Networks

LSTM: Long Short-Term Memory

ROC: Receiver Operating Characteristic

Spam: Unsolicited Bulk Messages

TN: True Negative

TP: True Positive

LR: Logistic Regression

GRU: Gated Recurrent Unit

ISH: Image Spam Hunter

UBE: Unsolicited Bulk Email

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Spam emails are unsolicited message content shared through emails to several recipients using electronic devices (Sharaff *et al.*, 2016). In most cases, cybercriminal have no existing relationship with the targeted recipient and collected their contact link from multiple sources including phone books, spam messages and tagged filled forms before sending the spam mails for malicious purposes. Over the last decade, email has become inundated with spam content. Image spamming is a new tactic invented by the spammers to attach an unsolicited image content in a binary format with a textual based content to avoid detection by text-based spam filters.

The widespread availability and significant increase in the use of the Internet has facilitated a fast and simple method for online transaction as well as various types of online communication, the most popular of which is emailing. Therefore, sending and receiving emails as a primary mode of communication has become very common (Shams & Mercer, 2016).

Email is almost a requirement for e-transactions. Despite the availability of various types of e-communications, sending and receiving e-mails has maintained its position as the simplest and fastest method of e-communication. It is one of the most widely used, fastest, and most efficient methods of exchanging information. However, due to the widespread use of email, there has been an increase in the number of problems caused by Unsolicited Bulk Email (UBE) messages, which is also known as email spam. Nevertheless, emails have remained successful in the field of online business transactions and are now required for other forms of online communication (Samira *et al.*, 2017).

The increased use of email applications and online transactions via email has resulted in to a very high percentage of email spamming which has relatively become a critical problem in the area of computing. Spamming as a rapidly increasing type of cyber attack, it is the most critical threat to every email users along with warms, viruses and phishing (Sharaff *et al.,* 2016).

According to Statista website (Statista, 2019), the world wide spam volume as per the overall e-mail traffic percentage, sorted by month and most recently reported period globally, was approximately 53.5% of the entire email traffic since September, 2018. During the second quarter, 2018, China was behind and responsible for the most of email spam contents, accounting for 4.36% of global email spam frequency.

The increasing rate of email spam is continuous and alarming, It has created a major problem for service providers, jeopardizing user confidentiality and causing resource loss. Because they cause enormous losses for organizations, ranging from the loading of mail server, and waste of bandwidth to the client profitability because of the period of time spent while detecting and dealing with the email spammers.

Email spam contents are used for a variety of cybercrime and to circumvent the security measures not only to increase device functionality and loss of storage facility. This violence has the potential to be employed for abusing client data integrity and to steal a percentage of sensitive information including financial information and passwords. The high volume of spam mail that circulates among networked computers has a negative effect on the memory space of the emil server, bandwidth management, the system processing power and application time. Spam email is becoming more of a threat on an annual basis, accounting for more than 77 percent of all email traffic globally (Fonseca *et al.,* 2016).

There have been numerous machine learning techniques for identifying unwanted spam. Despite the relative improvements achieve in the number of existing literature reviewed, there is no classification technique that has achieved 100 percent accuracy (Chopra *et al.*, 2015). For classification, each algorithm makes use of a limited set of features and properties.

Because of the important role of splitting ham (non-spam) from spam emails, two approaches have emerged: Knowledge engineering, which uses rules to separate spam from ham email and Content based engineering, which is a machine learning technique that takes decision based on the information similarities and uses heuristic approach to extract ham from spam email through learning from email traffic and then, training the rest corresponding email for the procedure to remain even with no further training (Fonseca *et al.*, 2016).

The majorities of current spam filters are only capable of identifying text based spam or image spam. In this research, a multi-modal architecture is developed that can identify textual spam, image spam, and mixed spam.

1.2 Statement of the Research Problem

Emailing has become a quick and cost-effective method of distributing business and personal details in a manner that is convenient to both the sender and the receiver. However, because of the conveniences and its simplicity of use, it has also become a hub of scams. There is a frequent overflowing of unsolicited emails in our inbox. As a result, determining the best classifier with reliable tools to detect spam and ham mails separately has become necessary. Distribution of email spam content over the Internet through emails has become a major challenge in the area of computing. Since last decade, emails are flooded with unwanted content such as sexual, marketing and other inappropriate content attached within the images known as spam images. This study is

limited to the classification of textual and image based spam using multi modal features, specifically using Recurrent and Convolutional Neural Networks, with the goal of providing better results compare to the previous papers on the same classification probelm.

Most of the current spam filters are single modal techniques which are limited to detecting specifically either textual spam or image spam and no classification technique has achieved 100 percent accuracy (Chopra *et al*,. 2015). In this work, a multi-modal architecture capable of detecting textual spam, image spam and mixed spam was developed.

1.3 Aim and Objectives of the Study

The aim of this research is to develop a spam detection techniques using multi-modal architecture.

The objectives of the research are to:

(i) Build a classification model for textual based spam

(ii) Build a classification model for image based spam

(iii) Combine the models in (i) and (ii) above based on multi-modal fusion.

(iv) Evaluate the performance of the model using performance metrics such as: Accuracy, Recall, F1-score and Precision.

1.4 Scope of the Study

For the purpose of clarity, the scope of this research is to provide an approach that is capable of handling both text and image spam content by considering their properties to classify and separating them from legitimate emails. This research is limited to conducting a spam classification using multi-modal architecture.

1.5 Significance of the Study

Spam content abounds on the Internet either in the form of text or unsolicited embedded in images. Although prior strategies for identifying textual spam have proven to be effective, spammers are continuously devising new ways to deceive them. Image spam is one such complex challenge, and it is the focus of this study. The outcomes of employing neural networks and deep learning to classify spam photos are discussed in this study. There has been limited deep learning research in this subject since its inception. By utilizing such strategies, emailing system users can be able to eliminate spam content including those embedded in images.

1.6 Justification for the Study

This work is intended to further make improvements on the design of email spam classification model from the Machine learning methods studied to filter and reduce spam email penetration using existing techniques.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Categories of Spam

Spam detection techniques are broadly classified into two categories:

2.1.1 Content-based spam

This type of spam includes text-based spam in emails. In this case, classifiers deal with the email's actual content, which is derived from email headers, keywords, and the email body, among other things. For spam classification, a wide range of machine learning approaches are available. It is now normal practice for email users to hide the client header, yet, this is the reason why the majority of people are unable to read their email header. As a result, the header is created simultaneously with the email's content (Abdullahi *et al.*, 2021). E-mail messages are frequently used as a choice between displaying or not displaying e-mail. The main idea behind this method is to figure out which part of the email course is being squandered. The email header contains a variety of fields that provide valuable information margins (Rusland *et al.*, 2017).

2.1.2 Non-content based spam

These include sophisticated kinds of mail spam such as image spam. The image attributes can be utilized to classify image spam, or with the advent of deep learning, images in raw byte form can be used. Image spams are classified into three generations, from the first to the third. Images in the first generation are just spam, however, images in the second and third generations are hidden with noise and background images to make them resistant to Optical Character Recognition (OCR) system. OCR algorithms can extract the portion of an image containing a given object for further processing, such as text extraction or object detection.

These approaches are typically used to extract text from photos using an OCR program. OCR (optical character recognition) is a mechanical or electronic description of certified images

physically typed, typewritten, or printed using machinery decoded information. It is frequently used to convert general ledger into database records for use in an institute's existing record keeping model or to disseminate the site's file. OCR can recognize and edit text, search for words and phrases, save, display, or create a replica that is free of scanning artifacts, and afterwards institutions such as machine analysis, text mining, and spoken text.

2.2 The Approach to Spam Detections

Foqaha and Mohammed (2016) shows that machine learning technique has gotten a lot of attention, and there are many other approaches that can be used in e-mail filtering. Among these are Nave Bayes, support vector machines, Artificial Neural network, K-nearest neighbor, Rough sets, and the artificial immune system. At the moment, there are two general approaches to spam filtering or email filtering. Two examples are knowledge engineering and content-based filtering in e-mail classification. To classify emails as spam or ham, the knowledge engineering technique must include a set of rules. The user, the filter, or another authority (such as the software business that delivers a certain service) should create a set of such rules. Because the rules must be changed and maintained on a regular basis, this technique yields no promising outcomes because it is not a real-time procedure and is inconvenient for most users.

2.2.1 Content based filtering

Content-based Filtering is a Machine Learning technique that makes decisions based on information similarities. This technique is commonly employed in recommender systems, which are algorithms that advertise or recommend goods to consumers based on information obtained about the user. These methods are based on feature extraction and image content analysis. These filters are used to evaluate and analyze the image's content and processes. The machine learning technique has gotten a lot of attention, and there are many other approaches that can be used in e-mail filtering. Among these are Nave Bayes, support vector machines, Artificial Neural network, K-nearest neighbor, Rough sets, and the artificial

immune system (Foqaha & Mohammed, 2016). At the moment, there are two general approaches to spam filtering or email filtering. Two examples are knowledge engineering and content-based filtering in e-mail classification. To classify emails as spam or ham, the knowledge engineering technique must include a set of rules such as the user, the filter (the business software that delivers a certain service).

2.2.2 Knowledge engineering approach

Masoumeh and Seeja (2015) indicates that knowledge engineering and machine learning are two general approaches used in e-mail filtering. The machine learning approach is more efficient than the knowledge engineering approach since it does not require the specification of any rules. Instead, a collection of training samples is used, with these samples consisting of post e-mail content. The categorisation rules are subsequently learned from these e-mail messages using a specific algorithm. Blacklisting, detecting bulk emails, grey listing, and scanning message headings are all part of the knowledge engineering strategy (Masoumeh & Seeja, 2015).

i. **Blacklisting** is a technique for detecting unique IP addresses that send a lot of spam. These Port numbers are added to a Domain Registration System-Based Black Hole List, and emails from those IP addresses are denied in the future. Spammers, on the other hand, employ a larger number of IP addresses to avoid these listings.

ii. **Detecting bulk entails** is yet another method for detecting spam. This method determines whether or not an email is spam by counting the number of recipients. Many legitimate emails, on the other hand, can receive a lot of traffic.

iii. **Scanning message headings** is a fairly accurate method of detecting spam. Spammers write programs that produce email subject lines. These headings may include mistakes that prevent them from complying with standard heading requirements. The presence of mistakes in these headings indicates that the email is

most likely spam. Spammers, on the other hand, are learning from their mistakes and making them less frequently.

iv. **Greylisting** is a technique for rejecting emails and returning an error notice to the sender. Spam filters will disregard this and refuse to resend the email, whereas people are more likely to do so. This technique, however, is inconvenient for humans and is not a perfect solution.

2.3 Machine Learning Definition

According to (Borde *et al.*, 2017), machine learning is the application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The approach starts with observational or data, such as examples, actual experience, or education, in order to uncover patterns in data and make better decisions in the future based on the examples provided. The ultimate goal is for computers to learn on their own, without the need for human intervention, and to adjust their behavior as a result of that learning.

Machine learning is described as the process of discovering a target function (f) that best maps input variables (x) to output variable (Y),

$$Y=f(x) \tag{2.1}$$

This is a general learning task in which would like to make future predictions (Y) based on new examples of input variable (x).

There is also an error (e) that is unrelated to the data input (x).

$$Y=f(x)+e \tag{2.2}$$

This mistake could be caused by a lack of features that adequately characterize the optimum X to Y mapping. This error is characterized as fundamental error because it cannot be reduced no matter how proficient it become at estimating the target function (f). To put it another way, learning a function from data is a challenging problem, which is why there is machine learning and machine learning algorithms.

2.3.1 Machine learning methods

Machine learning algorithms take data elements and identify them as Yes or No. It's always a dual label situation. The machine is first trained by the human, who labels it Yes or No. Classification is the process of encoding and embedding data and deciding its labels as Yes or No on a consistent basis using machine learning approaches. Machine learning is a data analytics technique that helps computers to learn through expertise the same manner living things do. Instead than relying on a model based on a predetermined equation, machine learning techniques use computer techniques to extract information complex pattern recognition. As the number of samples available for learning increases, the performance of the algorithms changes. The machine learning technique does not require the establishment of any explicit rules," requiring instead a set of pre-classified samples. The categorization criteria are then taught using any suitable approach from this data (Borde *et al*., 2017).

2.3.2 Categories of machine learning

Machine learning algorithms are sometimes characterized as supervised learning methods based on the input and output data or signal and response learning nature.

2.3.2.1 *Supervised machine learning*

By applying what they've learnt in the past to fresh data, algorithms can anticipate future events using labeled instances (x,y). To predict future output values, the learning approach creates an inferred function based on a study of a known training dataset. After a sufficient amount of

training, the system may provide targets for any new input. Furthermore, the classification model may contrast its output to the desired output and find errors, allowing the model to be modified as needed. Supervisory machine learning algorithms encompass regression analysis, neural networks, and Support Vector Machines (SVM). It creates a statistical equation of a dataset that has both the desired inputs and outputs (Divya *et al.*, 2018). The data is referred to as training data. It solves classification and regression problems with labeled datasets, resulting in predictive modeling. Linear regression, support algorithms, logistic regression, naive Bayes, and K-Nearest Neighbor are some examples.

The Supervised Machine Learning system has two modes of operation: training and testing as described in figure 2.1 of the augmented machine learning diagram shown below.

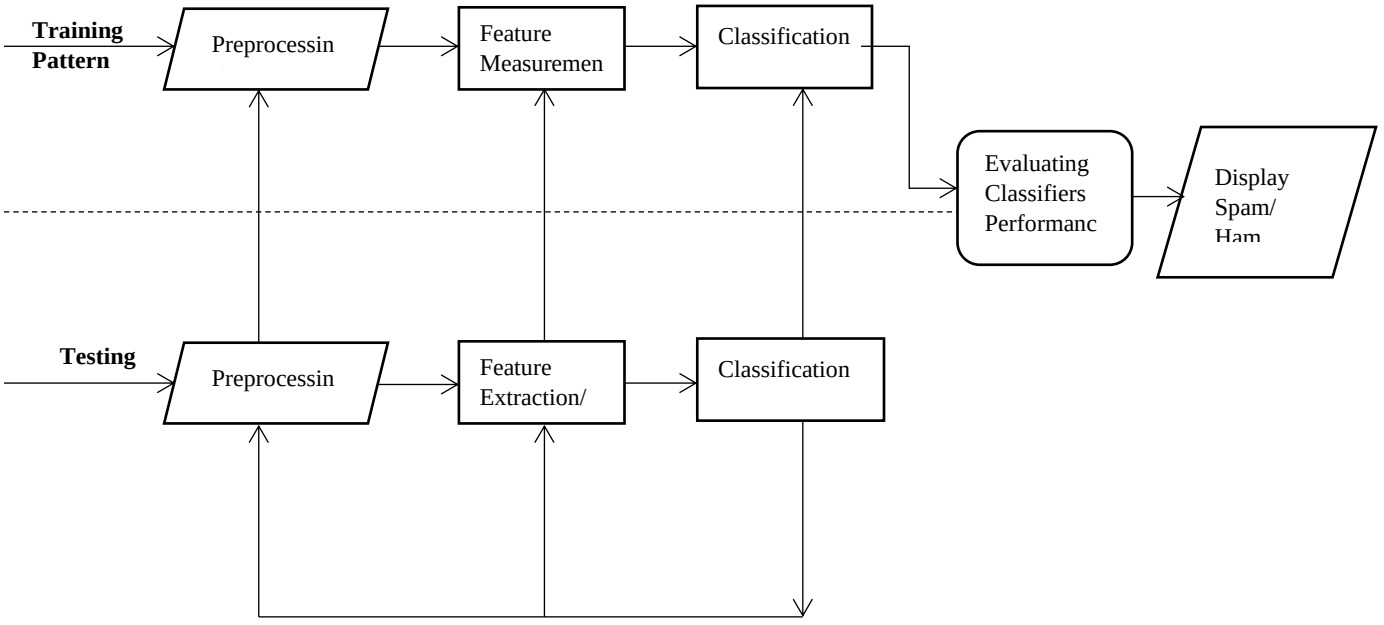

Figure 2.1: **A Model of supervised Machine Learning Algorithm Flowchart**

(Jayasingh *et al.*, 2016).

To produce and evaluate the results, the supervised machine learning classification algorithms go through the following stages: - Accessing and structuring the raw email dataset, then preparing this data and then, examining the data for analysis, and finally interpreting the

performance techniques. Figure 2.2 describes a simple design model for Machine Learning Data process chart flow.

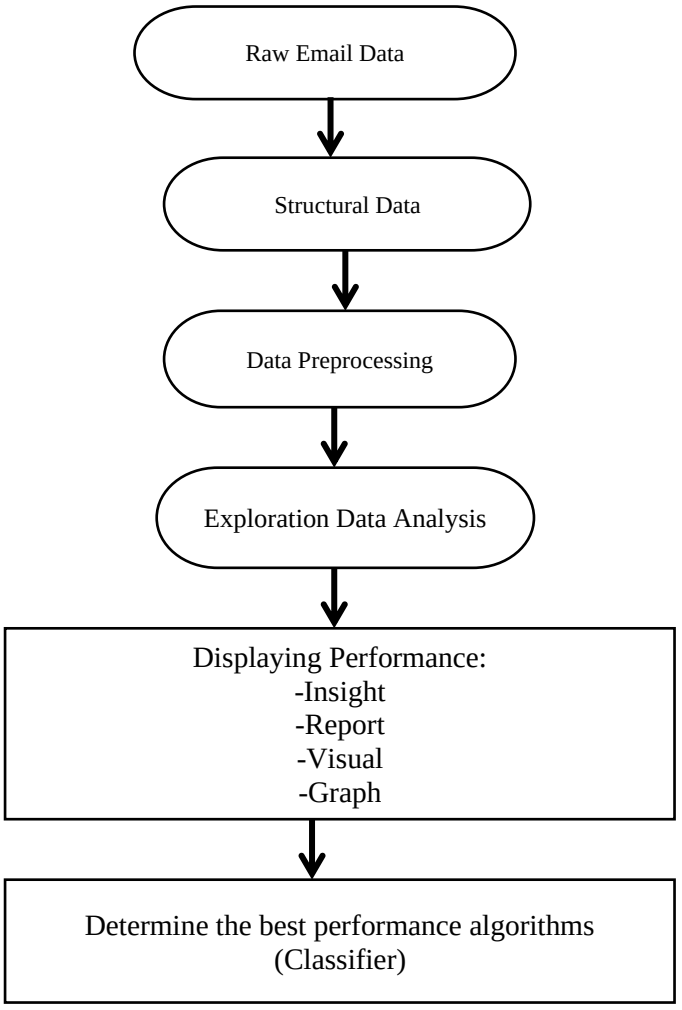

Figure 2.2 Structural Data flow diagram

(Swapna *et al*., 2017)

2.3.2.2 *Unsupervised Machine Learning*

Without Supervision, Machine Learning Algorithms are utilized when the training data is neither classed nor labeled (y). Unsupervised learning investigates how a system might infer a function from unlabeled input to describe a hidden structure (that is, an inherent structure). The system does not establish the proper output; instead, it analyzes the input and employs datasets to infer hidden patterns and relationships from large dataset.

These algorithms find structure in a set of data that only consists of inputs, such as data point grouping or clustering. As a result, unlabeled or unprocessed test data is used to train the algorithms. It is used to solve clustering (grouping) and anomaly detection difficulties. Unsupervised learning is used in the k-means clustering and association rule, making it descriptive modeling. Unsupervised Machine Learning Algorithms include: Anomaly detection utilizing the Density Function and Gaussian (normal) Distribution using K-means Clustering.

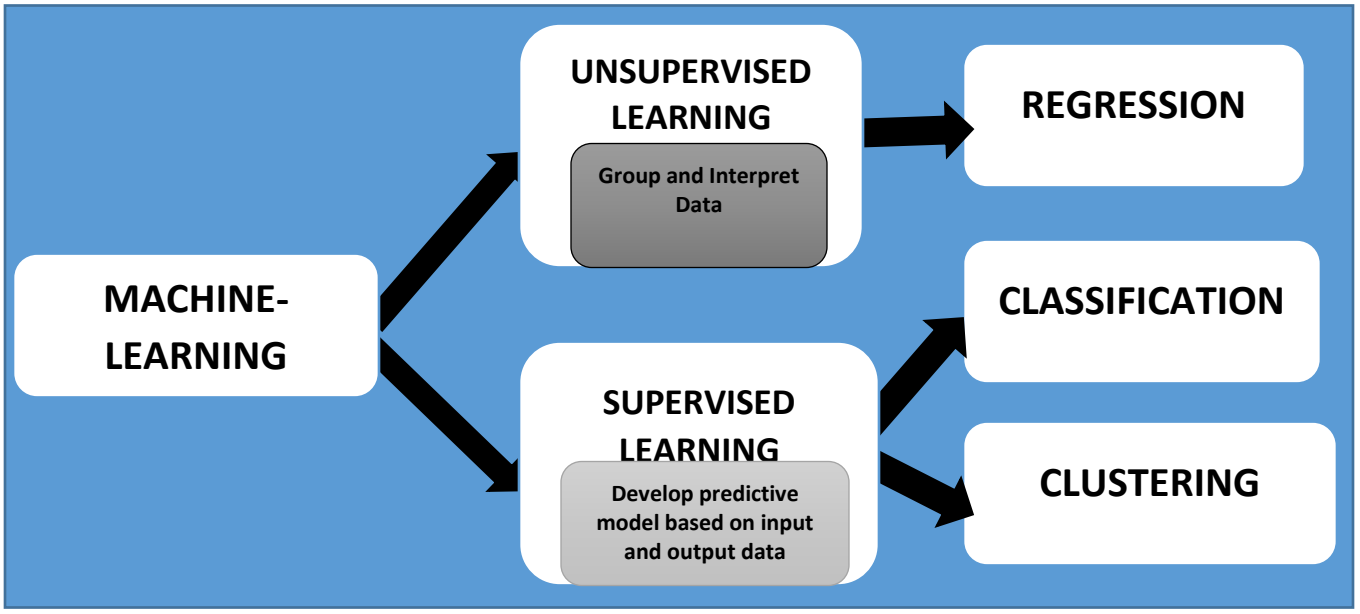

Figure 2.3 Machine learning Techniques

(Borde *et al.*, 2017)

2.4 Machine Learning Modeling

Our classification strategy is built around machine learning modeling. The decision to investigate multiple machine learning techniques for classifying both text and image emails as spam or not-spam in this step. Following that, the standard artificial intelligence techniques and deep learning methods are presented. It is discovered that a hybrid deep learning model based on a combination of two techniques Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) generated the best results when compared to the others based on the experimental findings shown by the methodology of this work.

2.4.1 Traditional machine learning

Despite the fact that the major goal of our method was to create a deep learning model, there is a need to discuss numerous algorithms from the classic machine learning family in this work, some of which are listed below:

2.4.1.1 *Naive bayes*

The classification process of this technique is an example of a learning techniques and also a predictive classification technique. According to Dipika and Kanchan (2017), naïve bayes is a fundamental probabilistic strategy for ethically capturing the clarity of a concept by measuring the likelihood of the outcome. It is used to solve analytical as well as quantitative problems. The Bayesian approach is named after Thomas Bayes, the researcher who invented it (1702-1761). Advanced information and analytical evidence can be integrated with classification to offer functional learning approaches. The Bayesian classification framework is useful for comprehending and assessing a wide range of learning methodologies. It finds the exact postulation potential and is resistant to noise in the supplied data. It's a straightforward probabilistic strategy based on Bayes analysis and based on solid assumptions that are self-contained.

2.4.1.2 *Clustering technique*

Clustering works by combining related pattern classes into a single group. Clustering is a type of method that breaks case studies into groups and compares them. This technique has attracted the interest of scientists and academics, and it has been employed in a variety of sectors. These are unsupervised learning approaches that were used to an email spam dataset with true labels, (Dipika & Kanchan, 2017). Given the availability of adequate representations, a variety of clustering approaches can categorize email spam datasets into spam or ham clusters as demonstrated in their work on email spam clustering.

2.4.1.3 *Support vector machine*

Support Vector Machine (CSVM) are a type of controlled learning algorithm that has been shown to outperform other learning algorithms. SVM refers to a group of algorithms that were developed to solve classification and regression problems. SVM has used application to provide answers to quadratic programming issues with inequality defects and sequential equality by dividing distinct classes using hyper plane. It makes the best possible use of the border (Dipika & Kanchan, 2017). Although the SVM is slower than other classification methods, it has a higher accuracy due to it ability to use a multifunctional model border that is not linear or sequential.

2.4.1.4 *Decision tree*

A Decision Tree (DT) is a type of classifier that uses a tree structure and follows a similar pattern to a decision tree. According to Dipika and Kanchan (2017), the decision tree classifier is a unique method that contributes to categorizing information. A decision tree node can be a leaf node that conveys the definition of the initial purpose (class) or a decision node that indicates that a particular verification should be performed with one branch and areas of work as a subset of the bigger tree that reflects any probable test outcome. Thus, decision tree learning is a technique that has been effectively used for filtering spam email.

2.4.1.5 *Random forest (RF)*

This is a popular instance of an ensemble learning technique that is suitable for classification of data in to classes (Dipika & Kanchan, 2017). For the first time, random forest was proposed by Torabi *et al.* (2015). The technique makes a specialized predictions using a tree structure. At the stage of training, some decision trees are created by the writer of the program. These decision trees are then applied for the task of predicting the group; this is done by considering the chosen groups of each tree and the category. These decision trees are then used for the

purpose of predicting the group, this is done by taking into consideration the selected groups in each tree and the group with the highest number of votes is taken as an output. Random forest approach is gaining more prominence these days and has been applied in a number of field and literature to solve the analogous problem according to Torabi *et al.* (2015).

2.4.2 Deep learning

In this paragraph, a description of the deep learning-based model for detecting spams is given. To implement this model, three different deep learning architectures are explored. To begin, the LSTM (Long Short-Term Memory) approach is used to design the architecture. Secondly, the CNN (Convolutional Neural Network) model is employed, and then a model that combined the two previous methods, the CNN and LSTM is created and implemented. The resultant model gave us improved performance results based on the experimental data.

2.4.2.1 *Recurrent neural network*

Stamp and Annadatha (2016) states that these are artificial neural networks that are used to recognize patterns in data sequences such as text, genomes, handwritings, spoken language, and numerical time series data.

Because RNNs have internal memory, they can remember important details about the input they receive, allowing them to be very precise in predicting what will happen next. This technique uses back propagation algorithms for training.

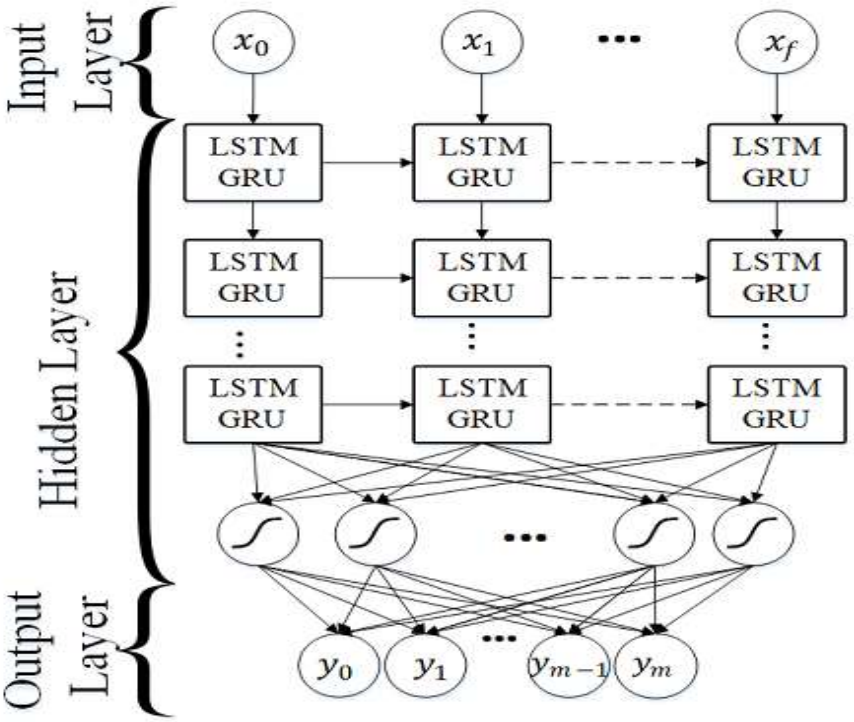

Figure. 2.4 General architecture of a RNN

Stamp and Annadatha (2016).

To prepare emails for the RNN model, a slightly different bag of words approach was used. As with the machine learning models, it is initiated by tokenizing the emails and creating a dictionary of all the words in all of the emails. Then, starting with 1, all of the words will be indexed. The collection of emails will then be iterated, with the dictionary being used to map each term in the email to the corresponding index. This method ensures that the email's word sequence is preserved. However, we now have a problem with each email's sequence length being different. To counteract this, we'll take the longest email and pad all of the others to that length. To avoid interfering with the training procedure, padding will be done by adding "0"s (zero) to the beginning of each sequence.

A gating method for RNN is the Gated Recurrent Unit (GRU). It is a simplified version of the LSTM design, with the following differences: GRU only has two gates and no internal memory as indicated in Figure 2.5.

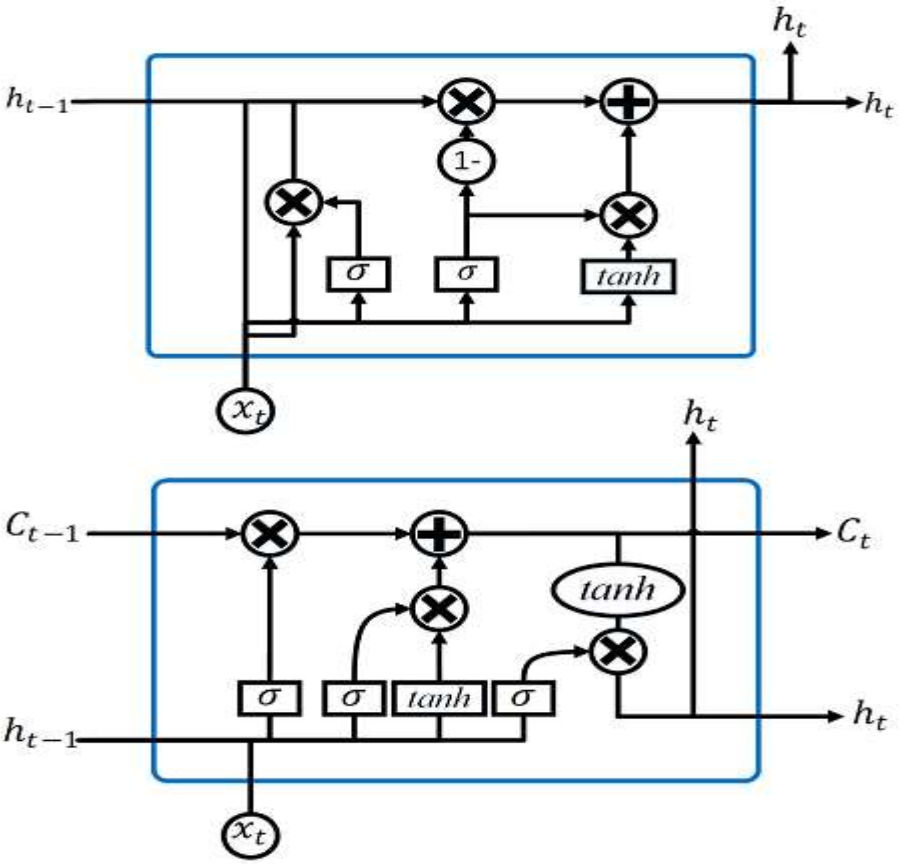

Figure. 2.5 Gated Recurrent Unit (GRU)

Stamp and Annadatha (2016).

Long Short Term Memory (LSTM) networks are a type of RNN that can learn long-term dependencies. These were created to circumvent the problem of long-term dependency that plagues most models. LSTMs help to simulate long-term interdependence by storing information over lengthy periods of time.

The model was developed using the Tensorflow framework. Several layers make up the LSTM model. Data will be passed through an embedded layer before being fed into the model. Each index that represents a word in a dictionary is represented by an embedded layer as a defined sized vector of random number values within a range. The relationship between the words can be modeled using this method. The embedded layer's output will then be routed through an LSTM layer. For this, a Tensorflows basic LSTM cell is used. The LSTM layer will include a dropout in its output to silence some neurons in the neural network at random, reducing the

risk of overfitting (Wadi *et al.*, 2017). The final output will be produced by a hidden layer that consumes the LSTM layer's output.

The logit (the end result that has not been sent through an activation function) will be provided to the cost function when it is calculated. 'The cost function utilized is softmax cross permeability with logits. The optimization function will then be used to lower the cost. The achieve ideal employed is the Adam optimizer.

The obtained logic value was passed through an activation function to make predictions. Because the task at hand involves a binary classification, the Sigmoid function was used.

2.4.2.2 *Convolution neural network*

Another deep learning architecture that can be used to classify hierarchical documents is convolutional neural networks (CNN). Despite being created for image processing with architecture similar to the visual cortex, CNNs have been effectively used for text categorization (Priyanka & Kare, 2015). In a simple CNN for image processing, an image tensor is convolved with a set of kernels with a size of d by d. These convolution layers are known as feature maps, and they can be stacked to provide various filters on the input. CNNs employ pooling to reduce computational complexity by reducing the size of the output from one layer to the next in the network. To lower outputs while keeping critical properties, various pooling strategies are applied.

The most common pooling method is max pooling, which selects the maximum element from the pooling window. To send the pooled output from stacked featured maps to the next layer, the maps are flattened into one column. The final layers of a CNN are typically fully connected dense layers. During the back-propagation step of a convolutional neural network, the feature detector filters and weights are updated. According to Sigma, the number of 'channels' in CNN for text is a potential issue (size of the feature space).

For text, this may be quite substantial for example 50K, but for images, this is less of an issue it has 3 channels of red, green and blue (RGB). This indicates that the CNN's dimensionality for text is really high.

Convolution Layers and Pooling Layers are connected in alternating way in a CNN, which is terminated by the Fully Connected Layer, which comes before the output layer.

(i) **Convolutional Layer**: The basic block of a convolutional neural network is the convolutional layer. The convolutional layer's primary function is to extract features.

(ii) **Pooling Layer:** To save processing time, a pooling layer is utilized to lower the dimensionality of the feature map. The pooling layer's main goal is to lower the spatial size (height, width). This reduces the number of parameters and, as a result, the number of computations. Pooling can be divided into three categories

(iii) **Fully Connected Layer**: This layer classifies the image using the features gathered by the preceding convolutional layers. Every neuron in the completely connected layer is coupled to every neuron in the previous layer. The output of the neural network is converted into probability for each class using the softmax function.

To use the CNN model for classification, a new deep network must be established, which necessitates training from scratch. To train from scratch, first decide the size of the photos that will be used as input, then configure the number of convolutional layers, pooling layer, and fully connected layer. Because a huge number of photos are necessary for the purpose, training time may be lengthy. For each new instance, a pre-trained CNN model is used. This model is always utilized because it has been trained on over a million photos and can categorize them.

A simplified representation of the CNN architecture is given in figure 2.6 below:

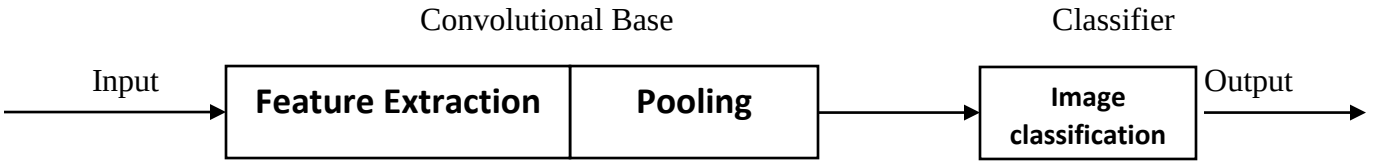

Figure. 2.6: General architecture of a CNN

Priyanka and Kare (2015).

2.5 Image Features

On characteristics derived from the image dataset, the first portion of the research used Recurrent Neural Networks (RNN) and Convolution Neural Networks (CNN). As described in prior work, this project employs 38 features. The features are divided into five categories: information, color, texture, shape, and noise. The distinct features that each category has are depicted in Table 2.1. The following are the different types of features:

2.5.1 Metadata properties

These parameters include the image file's height, width, aspect ratio, bit depth, and compression ratio (Priyanka & Kare, 2015). The compression of an image is defined as:

$$Compression\ Ratio = \frac{\text{height} * \text{width} * \text{channels}}{\text{file size}} \qquad (2.3)$$

2.5.2 Color properties

The average, skew, deviation, and probability readings of various picture variables such as RGB colors, kurtosis, hue, brightness, and saturation are examples of these qualities. Mean is a fundamental color feature that represents a picture's average pixel value. That is, it may be used to determine the background of an image. These features' histogram qualities differ between spam and ham images. Specific histograms show why specific color properties were chosen for classification.

2.5.3 Texture properties

The Local Binary Pattern (LBP) is a pattern that is used to make the comparison and information of surrounding pixels. The LBP appears to be an effective way for detecting picture spam, which primarily comprises of text on a white background. Instead of being dispersed, these scatter plots will be intense for specific values in spam images.

2.5.4 Shape properties

The Histogram of Oriented Gradient (HOG) analyzes how the intensity gradient of an image varies. HOG descriptors are good object classification descriptors since they are primarily utilized to express the structural shape and appearance of an object in an image. One of the most essential elements for detecting spam photographs is edges. They are used to draw attention to the image's limits. To find the edges, clever edge filters are utilized. The hog features for HAM and SPAM images are shown in Figures 2.7 and 2.8.

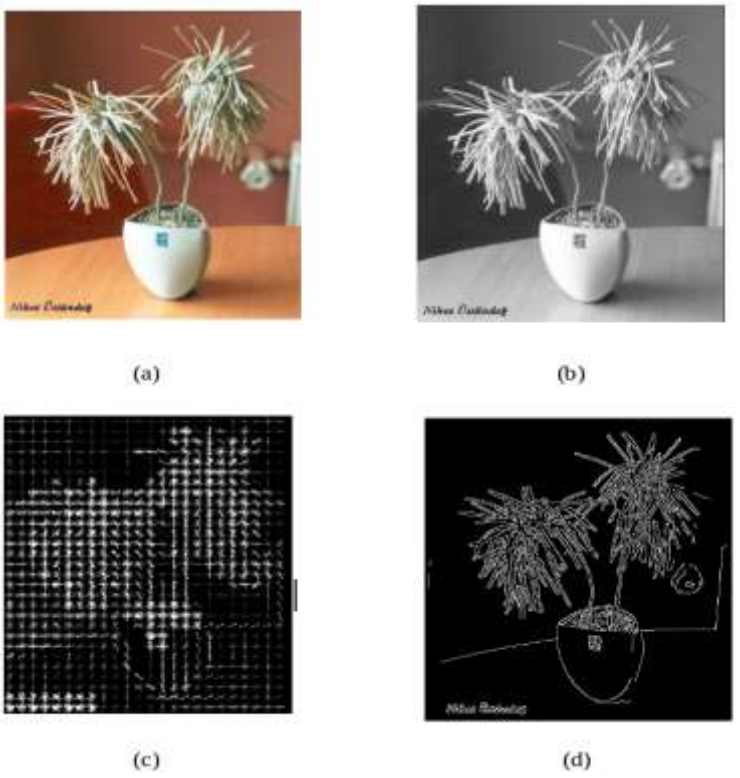

Figure 2.7: a) HAM original Image **b)** HAM Grayscale Image **c)** HAM HOG **d)** HAM canny-edges (Priyanka & Kare, 2015).

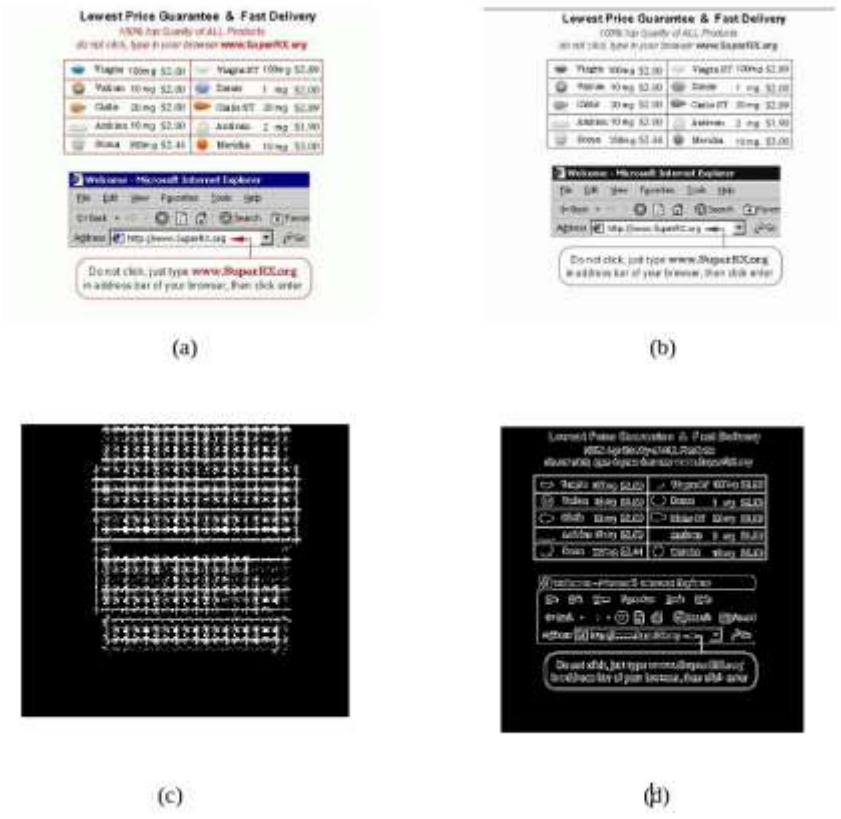

Figure 2.8: a) SPAM original Image **b)** SPAM Grayscale **c)** SPAM HOM **d)** SPAM canny edges (Priyanka & Kare, 2015).

2.5.5 Noisse propeerties

Signal to noise ratio (SNR) noise entropy are two of these characteristics. When compared to ham photos, spam photographs have less noise. The ratio of an image's mean to standard deviation is known as SNR.

Table 2.1 Different Image Feature (Priyanka & Kare, 2015).

Feature Type	Features	Description
i. Metadata	Height	Image Height
	Width	Image width
	Aspect ratio	Height and width ratio
	Compression ratio	Image compressed ratio
	File size	Image size on disk
	Image area	Image area
	Entr-color	Entropy for the color histogram
ii. Color	R-mean	Mean for red histogram
	G-mean	Mean for green histogram
	B-mean	Mean for blue histogram
	R-skew	Skew for red histogram
	R-var	Variance for red histogram
	R-kurt	Kurtosis for red histogram
iii. Texture	Lbp	Entry of LBP histogram
iv. Shape	Entr-hog	Entropy of HOG
	Edges	Total number of edges of an image
	Avg-edge-length	Average edge length
v. Noise	Snr	Signal to noise ratio
	Entr-noise	Noise entry

2.6 Machine Learning Based on Textual Email Spam Classification Approach

The study of Priyanka and Kare (2015) titled E-mail Spam Classification using Nave Bayesian Classifier uses the Lingapan database to classify spam and non-spam email, and there are many algorithms designed from this method that may be applied in e-mail filtering technique.

A feature representation method was utilized to train a Naive Bayes algorithm using the word-count methodology.

Abdulhamid *et al.* (2018) devised a performance analysis-based technique for email spam detection using classification techniques such as Bayesian logistic regression in their research on the quantitative comparison of classification techniques for email spam detection. Hidden Some of the ideas employed in this study include naive bayes, logic boost, rotation forest, neural network, logistic regression tree, rep tree, naive bayes, Support vector (RBF) network, voted recurrent neural networks, lazy bayesian rule, multifunctional support vector machine (svm:, random tree, and J48. These techniques' accuracy, precision, recall, F I-Measure, root mean squared error, receiver operator characteristics area, and root relative squared error were classified using the spam base dataset and Weka data mining tools. The rotating forest algorithms were known to have the highest F1-measure, whereas the nave bayes methods had the lowest F1-measure. They use the receiver operating characteristic (ROC) curves on randomly selected positive and negative instances, as well as the rotating forest algorithm, to determine the likelihood. The random tree received the lowest score of 0.90, while the ROC curves received the best score of 0.98. They also used kappa statistics to obtain the statistical results, and the output result for the rotating forest algorithm was significantly superior, with an approximate accuracy of 87.9%. The article found that rotation forest classifiers have the highest accuracy of 94.2 percent, followed by J48 (92.3 percent), naïve bayes (88.5 percent), and multilayer perception (93.2 percent).

Another work titled Classifying Unsolicited Bulk Email (UBE) using Python Machine Learning Techniques, Mohammed *et al.* (2016) proposed an approach for Classifying Unsolicited Bulk Email (U BE) using Python Machine Learning Techniques with the help of spam filtering which performs the work by creating a spam ham dictionary from the given training data and applying data mining algorithm to filter the training and testing data. After applying various

classifier on 1431 dataset, the approach predicts that. Naive Bays and SVM classifiers are the prominent classifier to spam classification.

Rathi and Pareek (2013) did a comparative performance analysis and proposed an approach for identifying the best classifier for email classification using data mining techniques in their research on spam email detection using data mining. They investigated a number of data mining methodologies for evaluating the effectiveness of numerous classifiers that used both a feature selection approach and one that did not. They considered the chosen algorithm for feature selection after deciding on the optimum feature selection approach. They explore with their data using a number of methods such as naive bayes, bayes net, support vector machine, decision tree, J48, random forest. There are 4601 occurrences and 58 attributes in the entire dataset. The results show that the random tree methods have the maximum accuracy of 99.72 percent and the naive bayes algorithm has the lowest accuracy of 78.94 percent.

Singh and Bhardwaj (2018) in their research on spam email detection using classification techniques and global training sets, they analyzed the solution and wanted to prove of spam filtering and recommended a combining classification strategy to achieve a superior spam detection outcome. The researchers employed a binary value system in which 1 represents spam email and 0 represents non-spam (ham) email. However, its success rate was low, so they used NB, KNN, SVM, artificial Neural Network classification methods to determine their accuracy, and then adopted a classification strategy for spam filtering based on the two methodologies of machine learning and knowledge engineering effectiveness. However, they collect data from a user training set, compare and discover spam emails, and then optimize the categorization algorithm using a global training set. The precision rate is increased by at least 2% when this technique is used.

Chopra *et al, (*2015) proposed a procedure of techniques for detecting malicious spam through feature extraction and improving the accuracy percentage and time for malicious spam detection techniques in their work on an approach for malicious spam detection in email with comparison between different classifiers. They measured the effectiveness of naïve bayes (NB) and support vector machine (SVM) classifiers based on computation time and accuracy reports. Preparing text data, building a word dictionary, extracting features, and training the classifier were all phases in this novel approach.

The authors separated the dataset into a training set of 702 messages and a test of 260 mails for text data processing, and then further classified into spam and ham mails. They then went through the feature selection procedure by creating a feature vector matrix. Among these techniques, the nave bayes was chosen as a better classifier.

Yuksel *et al.* (2017) developed a machine continuing to learn predictive analytics solution. The authors employed a machine learning approach and a decision tree to filter spam. Support vector machines were used as a supervised learning model to analyze data for spam categorization, whereas decision trees were used in data mining. The information was split into two sections: one for training and the other for testing. The algorithm was then trained and evaluated using Microsoft Azure's machine learning capabilities, and the results of the support vector machine and the decision tree technique were compared side by side. The SVM scored 97.6% after the analysis, compared to 82.6 percent for decision tree findings. The SVM classifier outperformed decision tree classifiers, according to these estimates.

Another paper, Voting-based categorization for e-mail spam detection by Al-Shboul *et al.* (2016), categorized 35 features into two groups: e-mail body characteristics and readability features. Using the Spam Assassin and CSDMC2010 datasets and three classifiers: Nearest Neighbor, Decision Trees, and Random Forest, they calculated Accuracy, Precision, Recall,

High False Positive Rate, and Fl-score for feature groups and the full feature set. These traits outperformed other studies that employed the identical email corpora and classifiers, according to the findings. They also reported the same assessment metrics after using the identical datasets and combining the three classifiers in various voting processes. They discovered that the results were superior than those produced by separate classifiers. The effect of introducing harmful features for boosting the evaluation outcomes of four classic spams was explored in the Journal of lCT Research and Applications.C4.5 decision trees, MLP, Naive Bayes, and Random forest are examples of email detection classifiers. They developed the four models utilizing the prior classifiers and the unbalanced data dataset Spam Assassin, which has 90 characteristics. These characteristics were divided into nine groups: Based on a header, a character, or a word Structural features, syntactic features Word and character content that is unique, Size, link-related properties, and attachment-related traits are all important considerations. According to three evaluation criteria, a comparison of the results before and after the addition of the detrimental features demonstrated that these factors significantly improved the classifiers' power to detect spam emails (Accuracy, Precision, and Recall).

Choudhary and Jain (2017) developed a novel strategy to detect and classify sms sparn utilizing a short message service (sms) system using machine learning classification algorithms. They begin with feature selection and the existence of mathematical terms such as ugls, dots, special symbols, emotions, lowercased and uppercased words, mobile number, keyword particular and SMS message length. Then they created a system that is designed to collect a dataset which contained 2608 emails out of 2408 collected sms spam corpus. The sms spam corpus v0.l comprise of two sets of messages as sms spam corpus v.0.1 small and sms spam corpus v.0l big. Using weka tools for five machine learning approaches. This includes logistic regression. naive bayes, J48, random forest, and decision tree. After the performance evaluation, the result for the following classification report: True positive rate (TP) and True negative rate (TN).

False positive rate (FP), False negative rate (FN), Precision, Recall, F-measure and receiver operating characteristics (ROC) area was obtained. The True positive rate and False Positive rate of the Random forest machine learning model algorithm was adjusted to performed better with high accuracy rate.

2.7 Machine Learning Based on Image E-mail Spam Classification Approach

The surge in spam emails has generated multiple comparison studies by neuroscientists on the performance of spam picture-based email classification algorithms employing hybridization measures, attracted the interest of the global academic community. As a result, determining which technique works best for a specific statistic is crucial in order to assure accurate email classification as spam or not. We provide an overarching structure and contemporary scientific research works in the literature with in scope of lowel-level approaches, such as Optical Character Recognition (OCR)-based techniques for filtering photo spam-based emails. and those that include the others techniques

Swapna *et al*. (2017) Use supervised machine learning methods to study spam email detection. The authors employed approaches such as inductive or supervised machine learning. This method uses algorithms to learn from a training dataset that includes both inputs and outputs, resulting in the production of a new model. After that, the new model is put to the test on new categorization samples. In the case of binary categorization, the output falls into one of two categories: spam or ham (that is a legitimate mail).

The authors then employed machine learning techniques such as neural networks, naive Bayes, support vector machines, lay algorithms, decision trees and artificial-immune systems to learn about incoming email behaviors and then classify them as spam or ham email based on their email datasets. During the review of this paper, a brief summary of the various methodologies was examined, and the various performance measures were assessed using the measure of

confusion matrix. The neural network was discovered to outperform the rest of the performance measures obtained.

Studies and work on the present trend in email spam by Alexy and Shyamanta (2016). They look at some of the most common spam traits, trends, and evasion strategies used by spammers, highlighting some interesting research approaches as well as some research gaps. These authors claim that filtering e-mail spam is challenging due to the dynamic behavioral character of spam; therefore, they advocate a thorough research of spam behavior to better understand the nature of spam and its evolution in order to build appropriate anti-spam countermeasures. A taxonomy of content-based email spam filtering, as well as a qualitative review of significant spam email surveys from 2004 to 2015 was carryout. After then, a report on new suggestions and findings of future investigations into machine learning strategies for emerging spam varieties was completed. The author next went over email corpus preprocessing, feature extraction, feature selection, and header and non-content feature analysis. The overview of the various spam filtering strategies utilized prior to machine learning was then spelled out, and the machine learning algorithm application to textual and multimedia content of spam emails was then adequately figured out. Recent methodologies that have emerged, as well as their conventional evaluation metrics, were given special consideration.

In the paper titled Improving Email Spam Detection Using Content Based Feature Engineering Approach (Wadi *et al.,* 2017), the authors designed a comprehensive and representative collection of spam email features using a very powerful and flexible feature extraction tool developed specifically for processing large email corpus. The produced dataset is used to train and test various classifilaction algorithms. This compare the performance of the four popular classifiers when trained based on extracting all features described in this work with the results obtained. The Knowledge-Based Spam Detection Methods: the effect of malicious 1 dated features in imbalance data distribution. The authors use three evaluation measures to evaluate

the developed spam detection mode1:-namely Accuracy rate, Precision and Recall to obtain the results in which the Random forest was noticed to produce the best classification results in the research.

Chopra *et al*. (2015) used a two-stage strategy to classify the textual section of a picture in order to determine whether the words in the letter were spam or not. The researchers noted in their study titled "The Textual and visual spam filtering" that spammers discovered a new method for embedding spam messages inside the image linked to the package, and that an OCR machine and a Bayesian technique were used in the initial phase. In an attempt to tackle this problem, the researchers are driven to propose a solution. A strategy was presented based on the hybridization of KNN and SVM. The primary idea is to categorize the nearby neighbors of a verification problem and generate a close by SVM for the task of segregation on the closed array.

Their work research was carried out utilizing the Dredze database and a public database, and the results demonstrate that the consistency as a performance metric has improved to roughly 98 percent.

Rusland *et al*. (2017) suggested in their work "An approach for image spam discovery and employing texture feature" a method for recognizing spam images by using image texture function. As one of the texture features in this investigation, the co-occurrence gray level matrix (GLCM) was used to each image. The photos are then identified using the feature that each image acquired. The Bayesian technique is used, as well as the neighbors classifier k-closest. The classifiers then use the 22 attributes collected to examine the photographs received from the Dredze and image Spam Hunter datasets To partition the datasets into training and test sets, cross validation methods are used. When compared to previous work, the results of their experiment, which included four measurement systems: validity, exactness, memory, and F-measures, show an improvement in this research domain and a significant reduction in

runtime; however, the study is largely limited to still deploying supervised learning.

Kumaresan *et al.* (2015) proposed a solution that removes particularly low-level features such as image metadata and histogram features. Due to the extracted features, a SVM classifier is applied with the aid of a function of kernel to detect image spam, the accuracy obtained with the method is 90% but their work is limited because of the time complexity. In this paper, they used multiple image features to build classifiers for image spam. The classifiers used are the combination of SVM and PSO. PSO improves the output by iteratively scanning candidate solutions and also ensure that the particle in the search space are moved. Again, due to its computational complexity, PSO is conveniently applicable only to the dataset that are relatively small as compared to SVM.

Wang and Kazuki (2014) proposed merging the attributes of spam images with the abundance of intersection points inside the images to filter spam photographs. The computation basic idea is that it uses the image percentage in the corner to determine whether or not this is spam. According to the researcher, most technological spam filtering solutions are useless for test messages hidden in images, which has been identified as a severe problem affecting the execution of transactions. The suggested solution made use of color feature extraction, image binarization, and corner point recognition. Furthermore, the testing findings of the suggested approach show that the detection performance of spam images is 90.5 percent. The analysis is carried out in an 8-bit RGB mode. The primary purpose of this experiment is to detect the corner and undertake a scientific calculations; nevertheless, this technique has the disadvantage of being unable to deal with sophisticated spam.

Meghali and Vijay (2014) suggested a method for identifying whether or not an embedded image is spam. The method is based on a hybridized interpretation of a single text section in an image. The technique is based on an interpretation of the image containing only one region of text and the dataset used is Dredze dataset, Classification methods are applied in a

hybridized manner. Particle Swarm Optimization is combined with Artificial Neural network for selection of features while the classifier for employed for spam classification and separation is Support Vector Machine. The learning ability of filters is the major strength of this method because every filter is different in terms of the data stored and model learned if every user receives different email but limited by complexity. The proposed framework is designed to handle both low level features and further processing of embedded text extraction. Their approach has been contrasted against other approaches and the result shows that AUC used in the proposed system for performance assessment is better than others methods.

Many conventional methods for detecting spam emails including the Bayesian method, the rule based system, Heuristic based filter IP blacklist, DNS black and white list holes applied a neural system strategy where neurons were trained and proposed an efficient techniques based on neural network for spam classification component to enhance the exactness, accuracy and F-review. The proposed system is contrasted with SVM and the result indicate that system is doing relatively better. The performance metrics used for the comparison are precision and accuracy. The approach of the plan is introduced to improve the accuracy quotient of the current methods (Sharma & Kaur, 2016).

Bhowmick and Hazarika (2016) machine learning algorithms were used to classify spam emails. The study focuses on spam filtering principles, activities, efficacy, and trends, as well as the most often used machine learning techniques to tackle the spam issue. Articles on spam image-based detection methods were looked for, reviewed, and chosen in ascending order of citation count. The benefits and drawbacks of existing machine learning algorithms for spam detection were discussed. The measured data of the selected spam image-based detection algorithms from the literature are tabulated, with an emphasis on four key performance indicators such as accuracy, precision, recall, and F-measure. The report

contains approximately 1000 spam phrases. The recommended approach, due to its overall average, can be combined with other algorithms to increase spam detection. In order to find the optimum strategy for email recognition and splitting, they evaluated a number of machine learning techniques in discrete mode on a dataset with 57 variables and a single target feature. The researchers analyzed the performance of several classificators. When the technique of feature selection was used throughout the testing, it was determined that the outcome showed accurate success. Random Tree was discovered to be the best spam mail identification classifier, with a 99.72 percent accuracy, and Decision Tree algorithm was discovered to become the second best classifier, with a 99.52 percent accuracy.

An overview of the reviewed machine learning approaches is provided based on the literature. Table 2.2 summarizes the summary in a tabular style, highlighting their advantages and disadvantages. The full summaries contain the study year, reference number, categorization methodologies, advantages, and limitations of each methodology.

Table 2.2: Summary for the classification Algorithms

Pub. Year	Techniques	Advantage(s)	Limitation(s)
2017	Naïve Bayes Classifier	Ethically handles ambiguity and influences the probability of result.	Depends on the assumption of Bayesian filter.
2016	Decision Tree	Has a very short training time.	Not flexible for adjustment.
2016	Random Forest	Provides higher performance and maintain lower classification error.	Longer training period.
2015	Support Vector Machine	Efficient mechanism during the data lost It has capacity to model un-sequential and un-straight forward borderlines which are not (multidimensional).	Slow classification.
2016	Clustering Technique	Capable of processing encrypted message content.	Cannot locate sensitive comparators.

Table 2.3 shows the result of the strategies in relation to the dataset used. The detail summaries include the publication year, reference number, dataset used, methodologies, Accuracy, Recall, F1-score and Precision in order to meet the review's goal.

Table 2.3 Summary of the classification Techniques with different performance metrics.

Year	Dataset	Techniques	Accuracy	Precision	Re- call	F- Measure
2015	Dredze	SVM	90%	-	-	-
2015	Spam Base	Naïve	84%	89%	78%	-
2015	ISH Dataset	KNN	93/7	97/96	91/0	94/35
		Naïve	99/1	98/50	98/5	99/25
2018	Spam base	Random	94.2	94.2%	94.2	94%
		Naïve	88.2	88.5%	88.5	88.5%
		Multilayer	93.2	93.3%	93.2	93%
		J48	92.3	92.3%	92.3	92.3%
2017	Dredze	Naïve	98%	-	-	-
2015	Spam base	SVM	79.5	79.02	68.6	-
		Naïve	76.2	70.59	72.0	-
2018	Spam base	ANN	92.41%	92.40%	92.4%	-
2018	ISH	Naïve Bayes	0.85	-	0.91	-
2016	Dredze	SVM	0.97	0.97	0.68	-
2018	ISH	J48	0.92	0.92	0.92	0.92
2017	Spam base	ID3-DT	0.89	-	0.90	-

CHAPTER THREE

3.0 RESEARCH METHODOLOGY

3.1 Machine Learning Mode of Operation

The Supervised Machine Learning system has two modes of operation: training and testing. In training mode, labeled data is provided to the machine learning system from a training data set. A huge number of emails are classified such as either spam or non-spam in this study's labeled training data (that is; ham). The classifier (the part of the machine learning system that does the actually predictions of future email labels) learns from the trained data to determine the links between the properties and labels of emails. During the testing mode where the machine learning system is fed with data which are not label. In this case, this data are the emails that have not been labeled as either spam or ham. Based on the email's properties, the classifier determines whether it is Spam or Ham.

3.2 The Methodological Design of Email Spam Analysis

To produce and evaluate the results, the supervised machine learning classification algorithms go through the following stages: - Accessing and structuring the raw email dataset, then preparing this data and then, examining the data for analysis (EDA), and finally interpreting the performance techniques to choose from. Figure 2.2 depicts a simple design model or Machine Learning Data process chart flow.

3.2.1 Datasets

In this research, we employ two publically available datasets that include text and images. All of these datasets contain spam and ham content derived from real email.

3.2.1.1 *Image spam hunter dataset*

Image spam Hunter developers gathered a big sample of image spam and a similarly large sample of ham images. This information is referred to as the ISH dataset. Following data

cleaning, 920 spam photos and 810 ham images from the ISH dataset were preserved for this study.

3.2.1.2 *Enron spam dataset*

The Enron dataset consists primarily of emails exchanged by Enron Corporation's senior management, Klimt and Yang (2004). This is the second dataset we used, and it is a publicly available dataset obtained From the Enron corpus where we choose only 20000 text emails for both the Spam and Ham on an equal ratio after discarding the duplicates and selecting randomly from the 33,645 text emails, of which 17108 emails are labeled as ham and 16537 emails are labeled as spam.

3.2.2 Data pre-processing

The Enron dataset archive was riddled with unnecessary files and corrupted data, making it impossible to extract features. In addition, there were several photos in the Image Spam Hunter dataset that had distorted content. To extract the initial frames and then saved using the appropriate format, all of these photographs were processed, which helped in the expansion of the datasets. Not all of the available information in an e-mail dataset is required or valuable. In most cases, removing less informative and noisy terms reduces feature space dimensionality and improves classification performance (Yuksel *et al.*, 2017). The act of turning the mail corpus into a consistent format that machine learning algorithms can understand is known as preprocessing. Due to the adversarial nature of spam, spam filters must constantly adapt to evolving spam strategies, notably in terms of feature extraction and feature selection. Whatever learning technique is used in order to have the content based filters trained, to either generate a private corpus or make use of a publically available corpus is very necessary. On the other hand, emails has to be preprocessed before they can be used to extract features. Furthermore, because a dataset may contain a huge number of features, it is necessary to carefully pick features to avoid classifiers over-fitting. Yes, it is right. Models with low bias that can learn

effectively from training data frequently have high variance and hence fail to generalize to new data, a phenomenon known as overfitting. Overfitting is characterized by a high model variance despite a low model bias (Yuksel *et al.*, 2017).

The efficacy and success of content-based spam fillers are dependent on feature engineering, which is the process of identifying and producing features that are more likely to increase the classifier's performance. The key phases in extracting features from an e-mail are as follows: Data cleansing, data integration, and data transformation are used to pre-process the email dataset.

3.2.3 Email spam classification

Email spam classification is a binary filtering task where by valid emails (Ham) get the short end of the stick (-), whereas spam gets the long end of the stick (+). A machine learning discipline is a field of computer science which investigates the design and development of computer systems that improve their performance automatically based on prior experience. Automatic e-mail categorization use statistical or machine-learning approaches to build a model or classifier specifically tailored to filter spam from a user's message stream. Machine learning approaches include the Naive Bayes Model, Support Vector Machine, Logistic Regression, Neural Network, Decision Tree and K-Nearest Neighbors are among the Machine Learning techniques used to combat spam. All of these models are part of the Supervised Machine Learning job. A set of pre-classified documents is required for the construction of the model or classifier (training set). The process of developing the model is known as training. All of these strategies have resulted in greater success for machine learning algorithms. Spam filters and detectors are based on Machine Learning's statistical basis.

As a result, training and developing a classifier on emails received by individual users is far easier than developing and refining a set of filtering rules.

3.3 Multi-Modal Architecture

In this work, a multimodal architecture for email spam detection is developed, in keeping with the study theme of multi-modal spam detection. To extract textual semantic relational characteristics and provide a classification probability value of the textual component of the email as spam, the LSTM model type of RNN is employed. And also build a CNN model to obtain the filtering probability result for the image segment as email spam. The resulting model will produce two classification outcomes.

The classification probability outcomes are then entered in to the last logistic regression model in order to obtain the ultimate filtering probability result which details the actual outcome of the email to be either spam or not. Dropout cogitation is used when an email contents is only textual based data or only image based data (Sharma & Kaur, 2016). In order to set the probability value of the LSTM model classification output or the probability value of the CNN model classification to be 0.5, an assurance that the resultant multimodal architecture is capable of handling not only textual or image based spam content but also the combination of the two which is the hybrid spam.

The new technique both text content and image content in a given email in allowing it to effectively filter out the spam content from the legitimate email whether in a separate mails or embedded in the image. This means that, the resultant methodology has the benefit of filtering not just a hybridized spam, but spam based exclusively on textual or image data. The output of the experiments reveal that the solution outperforms other strategies by a wide margin. The most significant change is such that, the CNN and RNN models are employed to detect and classify image and text data in an emails, and then apply the logistic regression method to hybridized them in to a resulting fusion model.

Spam filtering is a binary classification issue. This offer a spam filtering framework called Multi modal architecture to make our technique applicable not only for filtering mixed spam

data and also for detecting email spam only with text or image data. This structure is depicted in diagram form in Figure 3.1.

The following are the stages of this model for detecting spam:

i. Text and image preprocessing: Extracting both image and text data from available emails to create the dataset as input data.

ii. Finding the optimal classifiers: Both the text and image datasets are employed in training LSTM and CNN models, yielding the best LSTM and CNN models, respectively.

iii. Getting the probability values of the classification: To obtain the probability values of the classification. The text dataset is fed into the LSTM model while the image dataset is fed in the CNN model respectively.

iv. Getting the best resultant model: The probability values for the classification are supplied in to the multimodal fusion model which is then trained and optimized.

In the preceding descriptions, the probability value of the classification of a new email spam is obtain by following steps i, iii, and iv respectively.

Finally, the overall layout framework for the multi-modal architecture as well as the basic procedures in Figure 3.1

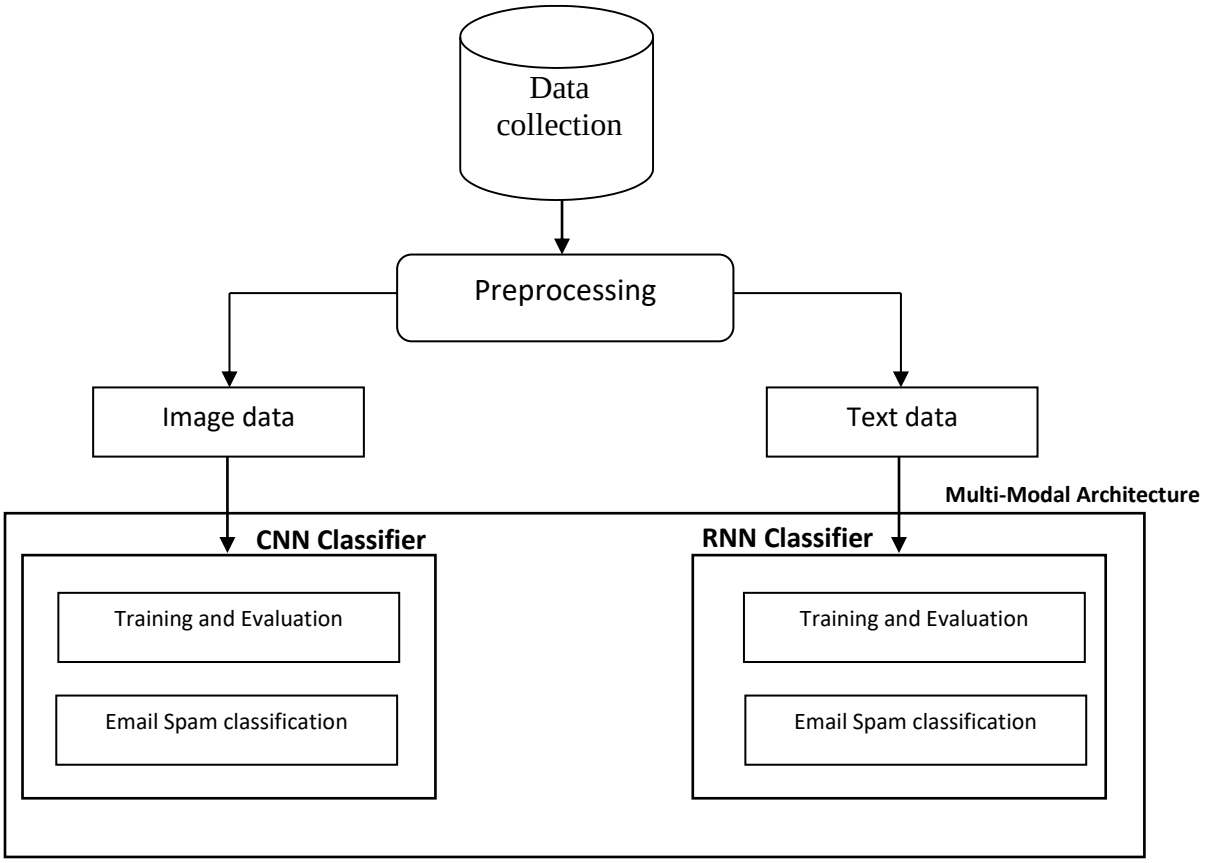

Figure 3.1 The Multi-modal Architecture

Training and testing were carried out on the planned model. This is done to assess the classification's efficacy. The standard classification performance metric accuracy is used to evaluate classifier performance. The most basic performance measure is accuracy, which is determine through the division of the well predicted observation to the total observation. This metric indicates the efficiency of the binary classification test works. What percentage of correctly predicted events actually occur. Accuracy alone isn't a good measure because it doesn't tell you how effectively the model detects positives and negatives independently. It assumes that the costs of both types of errors are equal. Depending on the difficulty, 99 percent accuracy might be outstanding, decent, middling, poor, or even terrible. Following preparation, each classifier is built using Python's Anaconda package and Keras with the Tensorflow backend.

3.3.1 Text classification model

In this phase of the work, designed and analyzed three architectures: LSTM, CNN, and a hybrid of CNN and LSTM.

The text classification model is made up of single word embedded layer, there are also two LSTM layers with a single fully connected layers (FC). The following ways are for handling the textual component of email spam in order to determine the probability value of the email spam classification: To get a word vector representation of an email, first capture its text data with the preprocessing technique, then utilize the word embedding technique.

After that, two LSTM layers were employed which have been set to automatically extract features from the text input. Lastly, the fully connected layer is used and an activation function known as the softmax activation function to determine whether the text data is a spam or ham.

The optimal probability values for text classification model of five hyper parameters such as: learning rate, batch size, epochs, dropout, and lastly, optimization procedure are determine through the use of grid search optimization techniques.

The LSTM model is described briefly in pseudocode here. Please consult the literature for a thorough LSTM unit algorithm (Yuksel *et al.*, 2017).

Let T stand for the email's text data.

Now, inputing T in to an embedding layer to produce a word vector given as:

$X, X = (X_1, X_2, \ldots, X_i)$,

Where $xi \epsilon R^n$ denotes n-dimension of word vectors for both i – th word of the T document.

Matrix $x \epsilon R^{i+n}$ represents T document.

While I is the maximum size of I<500.

The ct denotes memory

And ht denotes hidden state at time step t.

using the following equations:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \hat{c}_t \end{bmatrix} = \begin{bmatrix} s \\ s \\ s \\ \tanh \end{bmatrix} W \cdot [h_{t-1}, x_t], \quad (3.1)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \quad (3.2$$

$$h_t = o_t \odot \tanh(c_t), \quad (3.3)$$

where xt is the current input time.

f is the steps.

o are the forget gate activation, input gate activation and output gate activation respectively. t is the present cell state.

The LSTM model has been trained and optimized.

Algorithm 1 describes the full text spam classification algorithm procedure.

ALGORITHM 1: Algorithm for the classification of text email spam.

Input: Text Document (T).

Output: The classification probability value e for text spam.

i. Inputting T in to the word vector: X, X = (X_1, X_2, …, X_i),

ii. Input x at time t for the LSTM layer and complete the equations below:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \hat{c}_t \end{bmatrix} = \begin{bmatrix} s \\ s \\ s \\ \tanh \end{bmatrix} W \cdot [h_{t-1}, x_t],$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \quad (3.4)$$

$$h_t = o_t \odot \tanh(c_t),$$

iii. The text feature vector h = (h_1,h_2,,h_{64}) is obtained by the first LSTM layer;

iv. Input h at time t for the second LSTM layer and follow equations (1), (2) and (3) Finally, k = (k_l,k_2,,k_{32}) to obtain text feature vector k.

v. Apply the Softmax activation function to the FC layer to obtain the probability

value e of the text classification;

vi. **Return** e;

3.3.2 Image classification model

A CNN model is developed to classify emails in this portion of the research. The hyperparameters of the CNN model, the CNN architectures, as well as the architectures of the designed CNN image-based classifiers and the optimal value and range value these hyperparameters as obtained by the CNN model are all implemented.

The CNN model is developed to filter image based email spam specifically, in this portion of the research. the hyperparameters of the CNN model are shown in the architectural description below.

The CNN model has been trained and optimized.

Algorithm 2 describes the full imaget spam classification algorithm procedure.

ALGORITHM 2: Algorithm for the classification of image based spam.

Input: Image m, 128 x 128 RGB size.

Output: The probability value g for the classification of image based spam.

i. To the three convolutional layers, input m and you will obtain d, d= (d_1,d_2,,d_{64});
ii. To the first two FC layers, input d to obtain feature vector c, where c = (c_1,c_2,,c_{32});
iii. Input c to the last FC layer to acquire the probability value of categorization g using softmax activation function and that contains neurons.
iv. Return **g**;

3.3.3 LSTM-CNN (Multi-Modal)

Figure 3.3 depicts the structure of the multi-modal model. The goal is to obtain the most accurate classification probability value for email spam by combining the probability value for text classification and the probability value for image classification models and to obtain the resultant steps as follows:

i. To create a feature vector, combine the probability value for two LSTM and CNN classification models; $q,\ q \epsilon R^{1\times 4}$.

ii. To generate a comprehensive features vector by Inputting q in to the fully connected layers.

iii. Input the generated comprehensive feature vector in to the logistic layer.

iv. Inputting the comprehensive feature vector into the logistic layer.

v. Assume the probability dataset of the classification for the generated model to be as follows:

$D=\{(q_1, y_1), (q_2, y_2), \cdots, (q_v, y_v)\},\ qi \epsilon R^{l\times 4},\ y_i \epsilon\{0,1\},$

The logistic regression function's conditional probability distribution is as follows:

$$P(Y=1/q) = p(q) = \frac{e^{-w^T.q}}{1+e^{-w^T.q}} \quad (3.5)$$

$$P(Y=0/q) = 1-p(q) = \frac{1}{1+e^{-w^T.q}} \quad (3.6)$$

As the loss function, use the log-likelihood function, which has the following formula:

$$L(w) = \sum_{i=1}^{u} [y_i \log p(q_i) + (1-y_i)\log(1-p(q_i))] \quad (3.7)$$

$$= \sum_{i=1}^{u} [y_i \log \frac{p(q_i)}{1-p(q_i)} + \log(1-p(q_i))]$$

$$= \sum_{i=1}^{u} [y_i (w.q_i) - \log(1+e^{(w.q_i)})]$$

The Adam algorithm determines the greatest value of L(w). Furthermore, by optimizing L, the ideal estimated value for the parameter w can be derived (w).

If p>0.5, then the email is spam, else, it is a regular legitimate email.

3.4 Performance Metrics

Various performance indicators, such as Accuracy, Recall, F1-score and Precision, were employed to assess the usefulness of the suggested strategy. Furthermore, The Confusion Matrix, often known as the Error Matrix, is a popular tool for analyzing performance algorithm results. The confusion matrix is used as the primary evaluation metric for spam detection.

3.4.1 Confusion matrix (CM)

A Confusion matrix table is a table which is employed frequently to describe performance effectiveness of a classification model (classifiers) on a given test data which is also known as true data. It is a brief and logical categorization of task and prediction outcomes.

The confusion matrix is depicted in the Table 3.1, along with the anticipated column and actual class row names.

Table 3.1 A Confusion Matrix.

	Class1 **Predicted**	**Class 2** **Predicted**
Class1 Actual	**TP**	**FN**
Class2 Actual	**FP**	**FN**

The following components can be deduce from the definition of Confusion Matrix (CM):

i. True Positive (TP): The percentage of spam classified correctly.

ii. The number of correctly categorized legitimate emails in the True Negative (TN) category (Ham).

iii. The number of genuine emails that are incorrectly labeled as false positives (Ham) (FP).

iv. The number of spam messages that have been misclassified as false negatives (FN).

Each of the metrics described is applied to data so as to evaluate the performance measure and

compare the various classification reports in order to determine the best classifier model efficiency of the algorithm in the supervised machine learning method being used.

It is a feature of every good model to be able to successfully generalize in order to test data which is significantly different from the given training dataset.

A model built for training data by learning which scenarios fill the best may not perform well on test data. The confusion matrix is also calculated using the following parameters:

3.4.2 Classification accuracy

The relationship between classification rate and accuracy is:

$$Accuracy\ Rate\ (Acc) = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \qquad 3.8$$

Accuracy Rates: This metric indicates the efficiency of the binary classification test works. What are the percentage of correctly predicted events actually occur. Accuracy alone isn't a good measure because it doesn't tell you how effectively the model detects positives and negatives independently. It assumes that the costs of both types of errors are equal. Depending on the difficulty, 99 percent accuracy might be outstanding, decent, middling, poor, or even terrible.

3.4.3 Recall

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad 3.9$$

Rates of recall:- The recall is calculated ratio of the classified positive spam to the total positive spams. It describes how effective a test is at detecting positive spam. To put it another way, positive outcomes are predicted to be positive. The class was accurately identified if the recall was high (small number of FN). A good model should have a high recall rate. Sensitivity or TP Rate are other terms for recall. The recall (Rs) metric indicates how many spam messages the filter successfully prevented (i.e. the degree of its effectiveness of blocking actual spam correctly)

3.4.4 Precision

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad 3.10$$

The total number of properly categorized positive spam is divided with the total expected positive number of spam to determine the precision rates.. A high precision suggests that a test data email that has been flagged as spam is, in fact, spam (small number of FP Rate) the percentage of positive projections are correct. A high precision level is excellent. Precision (Ps) measures the proportion of messages labeled as spam by the filter that were, in fact, spam.

3.4.5 False positive rate

$$FP\ Rate = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \qquad 3.11$$

False Positive Rates (I-Specificity): This metric describes how well a model detects negatives. A high FP Rate is caused by a model that forecasts as positive when it is actually negative. This metric is sometimes given a one-star rating for specificity, which is defined as (TN Rare).

$$Specifity = \frac{\mathrm{TN}}{\mathrm{TP} + \mathrm{FP}} \qquad 3.12$$

It's preferable to have a high specificity (the ability to correctly forecast all negatives).

The bulk of positive spam is successfully identified.

High precision, low recall: This is an indication that much of positive spam denoted by (high FN) are lost.

3.4.6 F1-measure

Because there are two measurements (precision and recall), having a better measurement that encompasses both is advantageous. The F1-measure is calculated using Harmonic Mean rather than Arithmetic Mean since extreme values are penalized more severely. The smaller precision or recall value will always be closer to the F1-Measure. It is highly beneficial to have a high

f1_measures.

$$F1 - Measure = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad 3.13$$

CHAPTER FOUR

4.0 RESULT AND DISCUSSION

4.1 The Distribution of Datasets

As previously discussed, the results were presented and explained before delving more in details in to the network and then, displaying result obtained from the experimental results. Considering the raw image data from our data set and then, an alternative approach are demonstrated using raw images from our dataset and then, detail more on the result obtained from the LSTM-CNN architecture. The datasets discussed in Chapter 3 yielded all of the following results. On datasets, the LSTM and CNN models were trained and tested. The resulting model, on the other hand, was run on the combined dataset.

The table 4.1 describe the original email datasets distribution. The testing of 30% to 70% of training was carried out for all the datasets respectively. The result of this distribution is obtained as follows:

Table 4.1 Enron Email Dataset Distribution

EMAIL DATASET	SPAM	HAM	TOTAL
Enron Dataset Distribution	1000	10000	20000
Percentage Distribution (%)	50	50	100
	Testing data size of **30%**		
Training Distribution (70%)	7000	7000	14000
Testing Distribution (30%)	3000	30000	6000

The Enron spam dataset repository documentation.

Total email instances contained in the dataset was 33,645 text emails. After removing the duplicates, we selected 20000 text emails.

Spam email instance number: 16537 instances given from the Enron spam dataset repository

Ham email instance number: 17108 instances given from the Enron spam dataset repository.

Spam email instance percentage: 50% giving from the Enron spam dataset repository.

Ham email instance percentage: 50% giving from the Enron spam dataset repository.

4.2 The enron data size for training and testing

The Training and Testing datasets were run on the 20000 Enron spam dataset repository's Email instance.

The data size training of 70% and data size testing of 30% on 20000 produces

The Training Set = (70/100) * 20000 = 14000

Testing Set = (30/100) *20000 = 6000 . .

The Ham and Spam distribution of 14000 at 70% Training Set is as follows:

Training Set for Spam = (50/100) * 14000 = 7000

Training Set for Ham = (50/100) * 14000 = 7000

4.2.1 The data size testing of 30%

Both the Ham and Spam distribution of 6000 at 30% testing set as given below:

The Testing Set for Spam = (50/100) * 6000 = 3000

Testing Set for Ham = (60.6/100) * 600 = 3000

Table 4.2 Image Spam Hunter Email Dataset Distribution

EMAIL DATASET	SPAM	HAM	TOTAL
ISH Dataset Distribution	879	810	1689
Percentage Distribution (%)	50.8	49.2	100
	Testing data size of **30%**		
Training Distribution (70%)	601	582	1182
Testing Distribution (30%)	258	249	507

The ISH spam dataset repository documentation.

Total email instances contained in the dataset was 1739 images. After removing the duplicates, 1689 images were selected.

Spam email instance number: 879 instances given from the ISH spam dataset repository

Ham email instance number: 810 instances given from the ISH spam dataset repository.

Spam email instance percentage: 49.2% giving from the ISH spam dataset repository.

Ham email instance percentage: 50.8% giving from the ISH spam dataset repository.

4.3 The ISH Data Size for Training and Testing.

The Training and Testing datasets were run on the Email instance 1689 of the ISH spam dataset repository.

The data size training of 70% and data size testing of 30% on 1689 produces

The Training data set = (70/100) *1689 = 1182.3 approximately 1182

Testing Set = (30/100) *1689 = 506.7 approximately 507

The Ham and Spam distribution of 1182 at 70% Training Set is as follows:

Training Set for Spam = (50.8/100) * 1182 = 600.7 approximately 601

Training Set for Ham = (49.2/100) * 1182 = 581.54 approximately 582

4.3.1 Data size of 30% for testing

Ham and spam distribution of 507 at 30% set of testing data as given below:

The testing set for the spam = (50.8/100) * 507 = 257.56 approximately 258

Testing Set for Ham = (49.2/100) * 507 = 249.44 approximately 249

4.4 Model Performance

The evaluation results for both text and picture spam classification are provided in this subsection, as well as some analysis and discussion of the experimental data to evaluate the model performance for multiple fold cross validation. Table 4.3 shows the values of the measurement measures that were used.

Table 4.3 Model performance for text and image data

DATA TYPE	Accuracy	Recall	F1-score	Precision
Text Dataset	0.98	0.96	0.97	0.96
Image Dataset	0.98	0.97	0.97	0.96

It can be concluded from Table 4.3 that the multi-modal model developed in this research work has been successfully implemented for spam filtering; whether the email spam is contained text format or it is image based.

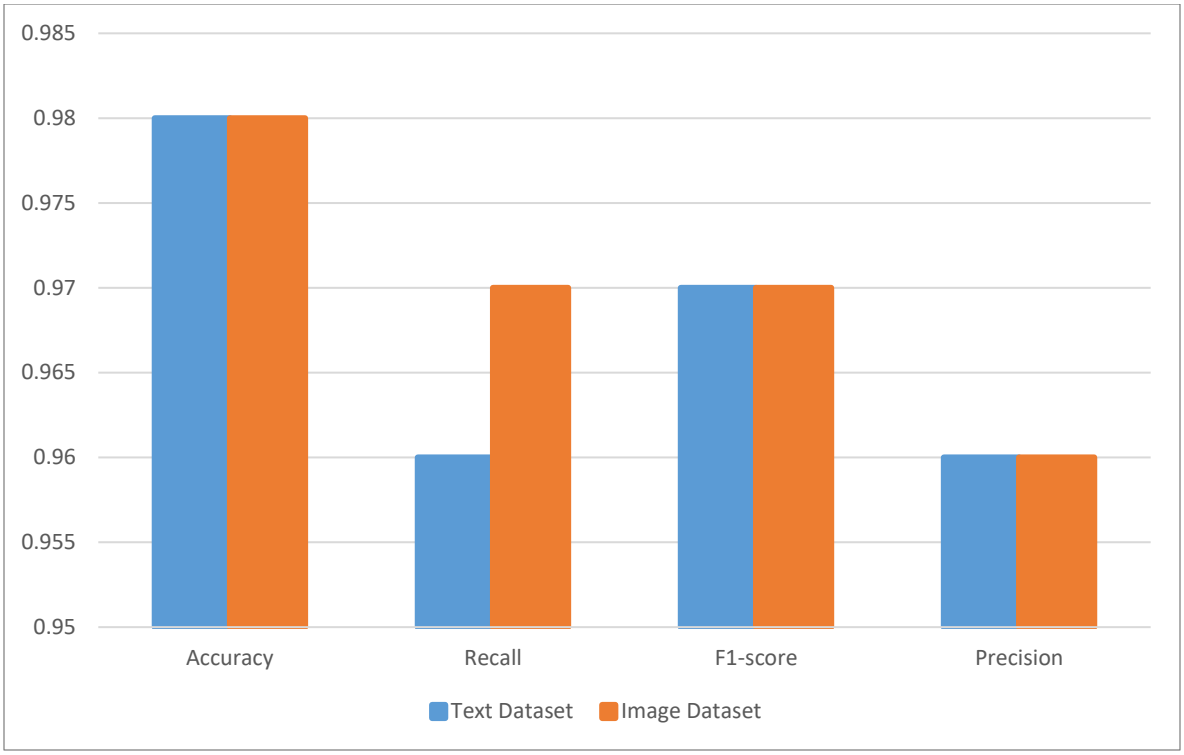

Figure 4.1 Model performance on text and image data

From Table 4.3 and figure 4.1, it has been established that the model performance on text and image datasets is significantly the same in terms of accuracy, F-score and precision but achieve different performance for Recall on text and image datasets.

Using a visual chart receiver operating characteristic (ROC) and confusion matrix to show and further illustrate the performance of the models. This can be thought of meaning the average sensitivity for the overall test value of the potential specificity.

The ROC chart is regarded to be an optimal analytical chart because of its Largest area Under the Curve (AUC). When the AUC value is obtain to be 0.5, then it shows no discrimination. (Meaning, the capacity to discern to whether or not an email is a spam at testing), between 0.7 to 0.8 is an acceptable indication, between 0.8 to 0.9 is denoted and represented to be excellent while, greater than 0.9 is given to be remarkable.

To further validate the model's performance, a comparative analysis to a large range of well performed existing models using the same or similar dataset. The textual dataset is used to test C-level CNN (Char-CNN), BiLSTM which is Bi-directional LSTM, and Immune Cross Regulation Model (ICRM), Naïve Bayes, ME, LSVM, CNN and ID3 Decision Tree are compared on the image dataset. The grid search optimization approach was used for the model to pick the optimal hyperparameters of SVM model, for example, are l and 0.00l, respectively, whereas the optimal hyper parameter k for the traditional K-NN model is given to be 1. Against compare our designed model's performance to a huge number of well-performing models utilizing the same or alternative dataset datasets. Tables 4.4 and 4.5 present the performance of the existing state of the art text and image spam categorization models:

Table 4.4 Performance of existing text spam classification models in their current state

Year	Dataset	Model	Accuracy	Precision	Re- call	F- Measure
2016	Enron	Char-CNN	0.96	-	-	-
2018	Enron	BiLSTM	0.964	-	-	-
2015	Enron	Naïve Bays	0.96	-	0.960	-
2017	Enron	ICRM	0.94	-	0.94	-
2019	SMS Spam	LSTM	0.91%	-	0.90	0.90
2016	Enron	Decision	96%	98%	94%	-

Table 4.5 The current state of image spam classification models' performance.

Year	Dataset	Techniques	Accuracy	Precision	Re- call	F- Measure
2015	Dredze	SVM	90%	-	-	-
2015	Spam Base	Naïve	84%	89%	78%	-
2015	ISH Dataset	KNN	93/7	97/96	91/0	94/35
		Naïve	99/1	98/50	98/5	99/25
2018	Spam base	Random	94.2	94.2%	94.2	94%
		Naïve	88.2	88.5%	88.5	88.5%
		Multilayer	93.2	93.3%	93.2	93%
		J48	92.3	92.3%	92.3	92.3%
2017	Dredze	Naïve	98%	-	-	-
2015	Spam base	SVM	79.5	79.02	68.6	-
		Naïve	76.2	70.59	72.0	-
2018	Spam base	ANN	92.41%	92.40%	92.4%	-
2018	ISH	Naïve Bayes	0.85	-	0.91	-
2016	Dredze	SVM	0.97	0.97	0.68	-
2018	ISH	J48	0.92	0.92	0.92	0.92
2017	Spam base	ID3-DT	0.89	-	0.90	-

4.4.1 Comparison of the Result with different models.

Table 4.6 Model performance for text and image data

DATA TYPE	Model	Accuracy	Recall	F1-score	Precision
	Char-CNN	0.95	-	-	-
	ICRM	0.94	-	0.95	-
Text Dataset	Naïve Bayes	0.96	-	0.96	-
	BiLSTM	0.95	-	-	-
	DT	0.96	0.98	0.94	-
	LSTM-CNN	**0.98**	**0.96**	**0.97**	**0.96**
	Naïve Bayes	0.85	-	0.91	-
	SVM	0.97	0.97	0.68	-
Image Dataset	Naïve Bayes	0.96	-	0.96	-
	J48	0.92	0.92	0.92	0.92
	ID3 D.T	0.89	-	0.90	-
	LSTM-CNN	**0.98**	**0.97**	**0.97**	**0.96**

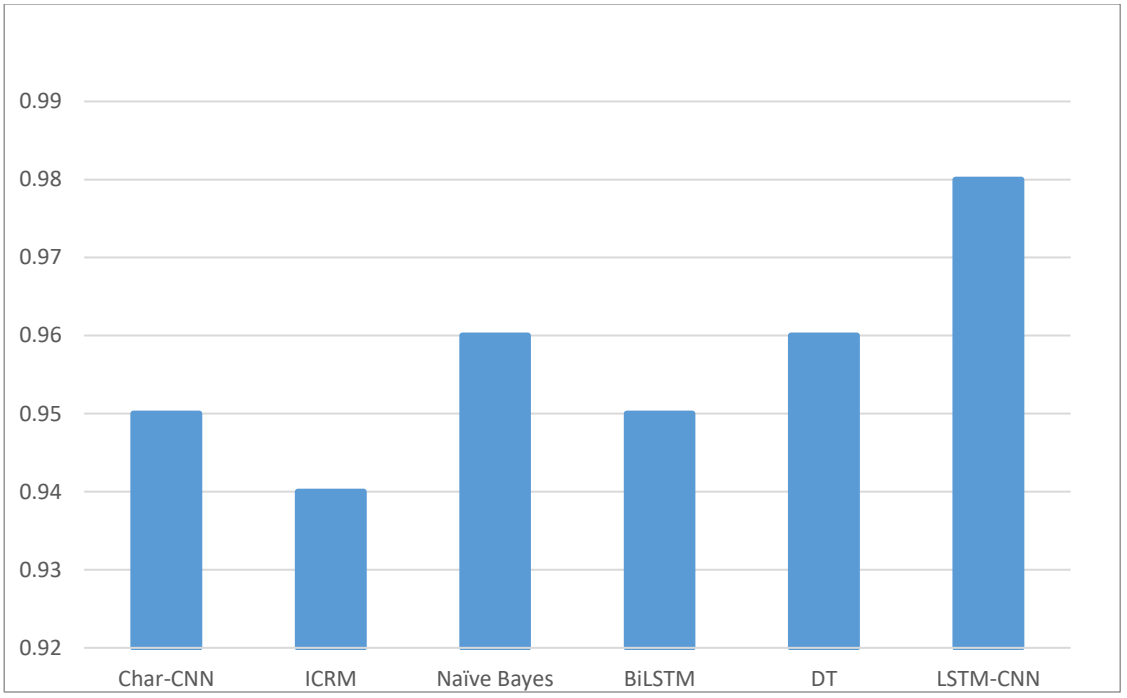

Figure 4.2 Performance comparison of the model accuracy on text datasets

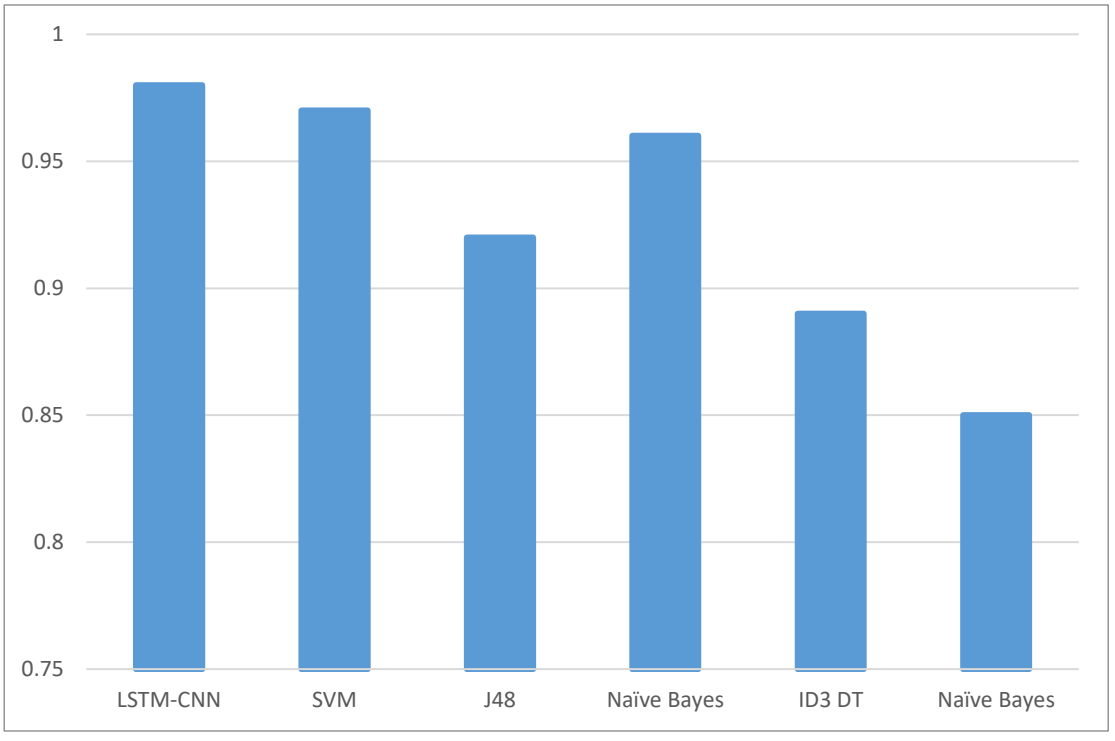

Figure 4.3 Performance comparison of the model accuracy on image datasets

For more clarity, Figure 4.2 and figure 4.3 above show the performance of the models based on the input data set, which can be either text or image data. According to Table 4.6, the LSTM-CNN model outperformed the other models on the text and image datasets.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Sending and receiving emails has remained the most convenient and time efficient method of online communication. The increase in online transactions via email has led to a significant increase in the global number of email spam which has relatively become a critical problem in the area of computing. There have been numerous machine learning technique for identifying unwanted email spam. Despite the significant improvements made in the number of existing literatures reviewed, there is no classification technique that has achieve 100% accuracy, each algorithm employs a limited number of features. As a result, determining the most appropriate technique is a critical task because their effectiveness needs to be weighed relative to their drawbacks. In order to improve spam detection rate, a multimodal architecture based on fusion technique is introduced by combining the Convolutional Neural Network (CNN) model and the Long Short-term Memory (LSTM) model through the use of logistic regression method. This is to implement a spam detection system capable of handling all the variety of email formats. The new model has the advantage of being able to filter hybrid spam as well as email spam containing either text data or image data, whereas prior model perform better only on handling text-based or image-based spam. The existing literature reviewed has indicated that a significant progress has been made email spam detection; however, more research effort needed improve on the performance and effectiveness of the multi-modal systems and to improve on the performance of artificial systems at large. More effort is also required to work more on the availability of well labeled dataset in order to enable a successful email spam classification. In this research, a variety of real-world text and image spam datasets have been utilized to develop powerful classifiers based on deep neural network architectures. In this work, a model is been produced with better outcome and which can learn even with an

increased dataset offered. However, despite the fact that it worked better, there is still potential for development.

5.2 Recommendations

i. In a real time applications, the dataset for spam detection exhibit a hug differences within the number of ham and spam emails. One-class filtering, adversarial generative network approach and short learning solutions are required to be proposed to handle imbalance between the positive and the negative training dataset samples.

ii. Due to the lack of a genuine publicly available mixed dataset for email spam classification, it is strongly suggested that a public mixed dataset be made available for effective and simple research purposes.

REFERENCES

Abdullahi, M., Bashir, S. A., Abdulmalik, D. M., & Abisoye, O. A. (2021). A Review on Machine Learning Techniques for Image Based Spam Email Detection. *2020 IEE 2nd International Conference on Cyber Space (CYBER NIGERIA),* 2(25), 120-125. doi:10.1109/CYBERNIGERIA51635.2021.9428826.

Abdulhamid, S. M., Osho, O., Ismaila, I., & Alhassan, J. K. (2018). Comparative Analysis of Classification Algorithms for Email Sparn Detection. *International Journal on Natural Language Computing (IJNLC),* 9(6), 111-123. doi:10.5815/ijcnis.2018.01.07

Alexy,Y. & Shyamanta, T. (2016). Approaches to E-Mail Spam Detection and Classification Using SVM and Feature Extraction. *International Journal on Natural Language Computing (IJNLC),* 2(4), 119-126. Retrieved from https://www.ijconline.edu/42689123.

Al-Shboul, A. B., Hakh, H. H., Faris, I. A., & Alsawalqah, H. (2016). Voting-based Classification for E-mail Spam Detection. *Journal of ICT Research and Applications,* 10(1), 29-42. doi:10.5614/itbj.ict.res.appl.2016.10.1.3

Bhowmick, A,. & Hazarika, S. M. (2016). Machine Learning for E-mail Spam Filtering Review, Techniques and Trends. *lnternational Journal of computer and Network Secuiry,* 4(2), 91–97. doi:10.48550/arXiv.1606.01042

Borde, S., Shinde, K. & Shaikc, B. (2017). A survey on Challegcs of Big Data Processing and Schedulling of processes using various Hadoop Schedulers. *lnternational Journal of Multifaceted and Multilingual Studies,* 3(1), 1-6. doi:10.4518/ijmmstu.41./l.2017.03.1.6

Chopra, Nisha, D., and Gaikwad, K. P. (2015). Image and text spam mail filtering. *International Journal of Computing Technology and Electronic Engineering (IJCTEE),* 5(3), 17-29. doi:10.1007/572-1109-10-9785-9_2

Choudhary, N., & Jain, A. K. (2017). Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique. *lnternational Journal* of *Advanced Informatics for Computing Research,* 712, 18 - 30. doi:10.1007/978-981-10-5780-9_2

Dipika, S., & Kanchan D. (2017). Spam E-mails Filtering Techniques, *lnternational Journal of Computer Technology Research,* 4 (6), 7–11. doi:10.1016/ijc.2017-05-10_2

Divya, K. S., Hargavi, P. B., & Yothi, P. J. (2018). Machine Laerning Algorithms in Big Data Analytics. *International Journal of Computer Sciences and Engineering,* 6(1), 63-70. doi:10.26438/ijcse/v6il.6370

Fonseca, D.M., Fazzion, O.H., Cunha, E., Guedes, P.D., Meira, W., & Chaves M. (2016). Measuring Characterizing, and Avoiding Spam Traffic Costs, *IEEE lnternational Journal of Internet Computing,* 20(4), 1-1. doi:10.1109/mic.2016.53

Foqaha, M., & Mohammed, A. (2016). Email Spam Classification Using Hybrid Approach of RBF Neural Network And Particle Swarm Optimization. *lnternational Journal of Computer of Network Security & Its Applications,* 8(4), 17-28. doi:10.5121/ijnsa.2016.8402

Jayasingh, B. B., Patra, M. R. & Mahesh, D. B. (2016). Security Issues and Challenges of Big Data Analytics and Visualization. 2nd *International Conference on Contemporary Computing and Informatics, 2(16),* 204-208. doi:10.1109/IC3I.2016.7917961

Klimt, B., & Yang, Y. (2004). The enron corpus: A new dataset for email classification research. *In European Conference paper on Machine Learning; Springer: Berlin/Heidelberg, Germany*, 3201, 217–226. doi:10.1007/978-3-540-30115-8_22

Kumaresan, T., Sanjushree S., Suhasini, K., & Palanisamy, C. (2015). Image Spam Filtering Using Support Vector Machine and Particle Swarm Optimization. *International Journal of Computer Applications*, 2(1), 17–21. Retrieved from https://www.ijcaonline.org/proceedings/nciprc2015/number1/20508-8006

Masoumeh, Z. & Seeja, R. K. (2015). Feature Extraction or Feature Selection for Text Classification: A Case study on Phishing Email Detection. *International Journal of Information Engineering and Electric Business*, 7(2), 60-65. doi:10.5815/ijieeb-2015.02.08

Meghali, D., & Vijay, P. (2014). Analysis of an Image Spam in Email Based on Content Analysis. *International Journal on Natural Language Computing (IJNLC)* 33313(3), 129-140. doi:10.5121/ijnlc.2014.3313

Mohammed, S., Mohammed, O., Fiaidhi, J., Fong, S. (2016). Classifying Unsolicited Bulk Email (UBE) Using Python Machine Learning Techniques. *lnternational Journal of Hybrid Information Technology, 6(1), 43-56.* Retrieved from https://www.academia.edu/24703712.

Priyanka, S., & Kare, P. (2015). E-mail Spam Classification Using Naive Bayesian Classifier. *international Journal of Advanced Research in Computer Engineering & Technology, 4(6),* 792-796. Retrieved from https://docplayer.net/23662591

Rathi, M., & Pareek, V. (2013). Spam mail detection. through data mining-A comparative performance analysis. *International Journal of Modern Education and Computer Science,* 5(12), 31-39. doi:10.5815/ijmecs.2013.12.05

Rusland, N. F., Wahid, N., Kasim, S. & Hafit, H. (2017). Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. *IOP Conference Series for Materials Science and Engineering,* 226(1), 12-91. doi:10.1088/1757-899x/226/i/012091

Samira, D., Mcryern, A., Bouabid, L. O. & Hicham, L. (2017). Towards A new Spam filtering Based on Paragraph Vector-Distributed Memory Approach. *The 4th lnternational Symposium on Emerging Inter-networks, Communication and Mobility,* 110(1), 486-491. doi:10.1016/ji.procs.2017.06.130

Shams, R. & Mercer, R. E. (2016). Supervised classification of spam emails with natural language stylometry. *Neural Computing and Applications*, 27(8), 2315-2331. doi:10.1007500521-015-2069-7

Sharaff, A., Nagwani, N., & Dhadse, A. (2016). Comparative Study of Classification Algorithms for Spam Email Detection. *Emerging Research in Computing, Information, Communication and Applications*, 2(23), 237-244. doi:10.1007/978-81-322-2553-9_23

Sharma, R., & Kaur, G. (2016) E-Mail Spam Detection Using SVM and RBF. *International Journal of Modern Education and Computer Science*, 8(4), 57–63. doi:10.5815/ijmecs.2016.04.07

Singh, V. K., & Bhardwaj, S. (2018). Spam Mail Detection Using Classification Techniques and Global Training Set. *lntelligent Computing and Information Communication*, 673, 623-632. doi:10.1007/978-981-107245-1-61

Stamp, M., & Annadatha, A. (2016). Image spam analysis and detection. *Journal of Computer Virology and Hacking Techniques*, 14(1), 39-52. doi:10.1007/511416-016-0287-X.

Statista. (2019). Global spam volume as percentage of total e-mail traffic from January 2014 to March 2019 by month. Retrieved from https://www.statista.com/statistics/420391/spam-email-traffic-share/.

Swapna, B., M, U., Viraj S, B., & Nilesh, M. (2017). Supervised Machine Learning techniques for Spam Email Detection. *lnternational Journal for Science and Advance Research III Technologyt' (JJSART), 3(3), 760-764.* Retrieved from https://ijsart.com/content/pdfDocuments/IJSARTV313867

Torabi, Z.S., Nadimi-Shahraki, M.H. & Nabiollahi, A. (2015). Efficient support ector machines for spam detection: a survey. *International Journal of Computer Science and Information Security*. 13(1), 11–28. https://www.researchgate.net/publication/316075352

Wadi, H., .Hossam, F., Ja'far, A., Ala', M. A., & Ibrahim, A. (2017). Improving Email Spam Detcclion USing ontent Based Feature Engineering Approach. *Jordan on Applied Electrical Engineering and Computing Technology, 2(1) 1-6.* doi:10.1109/AEECT.2017.8257764

Wang, J., & Kazuki, K. (2014). Image Content-Based" Email Spam Image Filtering. *Journal of Advances in Computer Networks,* 2(2), 110-114. doi:10.107763/JACN.2014.v2.92

Yuksel, A. S., Cankaya, S. F., & Uncu, L.S. (2017). Design of a Machine Learning Based Predictive Analytics System for Spam Problem. 3rd *International Conference on Computational and Experimental Sciences and Engineering*. 132(3) 500-504. doi:10.12693/Aphyspo1A.132.500

APPENDIX A. Importing packages and files:

```
import os

import numpy as np

from matplotlib import image, pyplot

from skimage.transform import resize

from keras.models import Sequential, Model

from keras.layers import Conv2D,MaxPool2D,Dense,Flatten,Dropout

from keras import callbacks

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, f1_score,

recall_score,classification_report

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.svm import SVC

from sklearn import svm

from sklearn.datasets import fetch_20newsgroups
```

APPENDIX B. For the model to determine whether the input data is a text or image, the following python code is used:

```
file_name, file_extension = os.path.splitext("/C:/Users/hp/Documents/SpamCollection.txt")
print(file_name)
print(file_extension)
print(os.path.splitext("/C:/Users/hp/Documents/.bashrc"))
print(os.path.splitext("/C:/Users/hp/Documents/SpamCollection/image.png"))
def is_binary(file_name):


   with open("SpamCollection", 'rb') as f:
      for block in f:
         if b'\0' in block:
            print('0')
      else: print('1')
```

The output:

```
/C:/Users/hp/Documents/mSpamCollection
.txt
('/C:/Users/hp/Documents/.bashrc', '')
('/C:/Users/hp/Documents/SpamCollection/image', '.png')
```

APPENDIX C. Input Data Display

```
file_name, file_extension = os.path.splitext("/C:/Users/hp/Documents/SpamCollection.txt")
print(file_name)
print(file_extension)
print(os.path.splitext("/C:/Users/hp/Documents/.bashrc"))
print(os.path.splitext("/C:/Users/hp/Documents/SpamCollection/image.png"))
def is_binary(file_name):
   with open("SpamCollection", 'rb') as f:
      for block in f:
         if b'\0' in block:
            print('0')
      else: print('1')
```

The output:

```
('/C:/Users/hp/Documents/SpamCollection/image', '.png')
.png
('/C:/Users/hp/Documents/.bashrc', '')
/C:/Users/hp/Documents/mSpamCollection
```

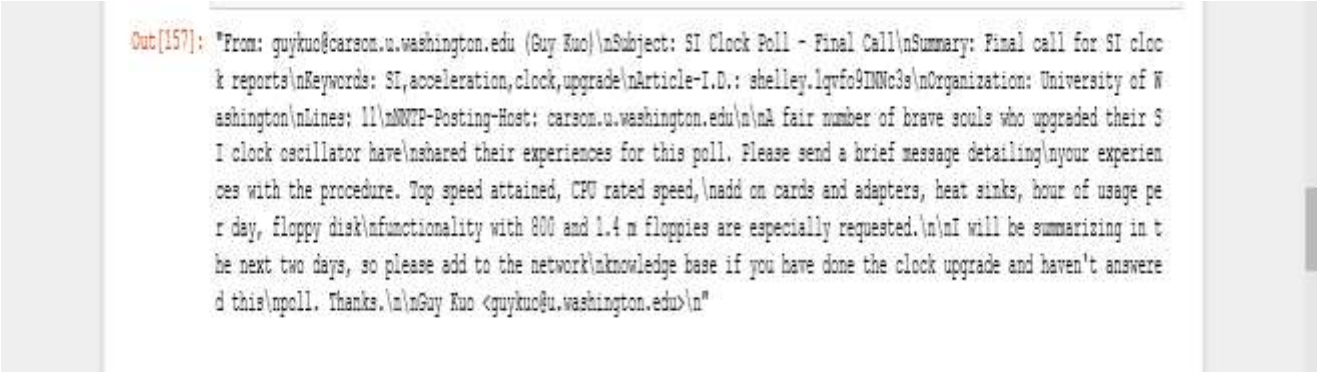

Figure 3.3 The visual display of the text datasets.

```
twenty_train = fetch_20newsgroups(subset='train', shuffle=True, download_if_missing=False)
twenty_train.data
```

APPENDIX D. The following is the Python code and result of the pre-processing of the imported spam email dataset by rescaling, standardizing and normalizing the data:

```
texts = twenty_train.data # Extract text
target = twenty_train.target # Extract target
# Load tools we need for preprocessing
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
vocab_size = 20000
tokenizer = Tokenizer(num_words=vocab_size) # Setup tokenizer
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts) # Generate sequences
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
# Create inverse index mapping numbers to words
inv_index = {v: k for k, v in tokenizer.word_index.items()}
# Print out text again
for w in sequences[1]:
   x = inv_index.get(w)
   print(x,end = ' ')
```

APPENDIX E. The following python codes read image data and their repositories:

```
for folder in os.listdir(DATA_PATH):
   if folder=="New_Spam":
     continue;
   print(">>>Reading ",folder)
   count+=1
for file in os.listdir(DATA_PATH+folder):
      if(str(file).endswith('.jpg') or str(file).endswith('.JPG') or str(file).endswith('.jpeg') or
str(file).endswith('.JPEG')):
         img = image.imread(DATA_PATH+folder+'/'+file)
```

```
        hsh = hash(tuple(np.array(img).flatten()))
        if(hsh not in hashList):
          spamData.append(resize(img, (156, 156, 3)))
          hashList.append(hsh)
          label.append(count)
spamData=np.array(spamData)
label=np.array(label)
```

The output:

```
>>>Reading  DATA_PATH
>>>Reading  NaturalImages
>>>Reading  SpamImages
```

To print the data shape:

```
print("Spam data shape : ",spamData.shape," Label shape : ",label.shape)
print("x_train shape : ",x_train.shape," y_train shape : ",y_train.shape)
print("x_test shape : ",x_test.shape," y_test shape : ",y_test.shape)
```

Output:

```
Spam data shape :  (898, 156, 156, 3)  Label shape :  (898,)
x_train shape :  (1194, 156, 156, 3)  y_train shape :  (1194,)
x_test shape :  (513, 156, 156, 3)  y_test shape :  (513,)
```

```
print("Number of train SPAM",len(y_train[y_train==0]))
print("Number of train HAM",len(y_train[y_train==1]))
print("Number of test SPAM",len(y_test[y_test==0]))
print("Number of test HAM",len(y_test[y_test==1]))
```

Output:

```
Number of train SPAM 627
Number of train HAM 567
Number of test SPAM 270
Number of test HAM 243
```

APPENDIX F. CNN model architecture

```
Model:
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 156, 156, 32)      896
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 78, 78, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 78, 78, 64)        18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 39, 39, 64)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 39, 39, 128)       73856
_________________________________________________________________
dropout_3 (Dropout)          (None, 39, 39, 128)       0
_________________________________________________________________
flatten_2 (Flatten)          (None, 194688)            0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               24920192
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 129
=================================================================
Total params: 25,013,569
Trainable params: 25,013,569
Non-trainable params: 0
```