

# Dynamic Spam Detection System and Most Relevant Features Identification Using Random Weight Network

by

Syed Mahbubuz Zaman

16201017

A. B. M. Abrar Haque

17101078

Mehedi Hassan Nayeem

17101261

Misbah Uddin Sagor

17101283

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
January 2021

© 2021. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:

*Syed Mahbubuz Zaman*

---

Syed Mahbubuz Zaman  
16201017

*A.B.M. Abrar Haque*

---

A.B.M. Abrar Haque  
17101078

*Mehedi Hassan Nayeem*

---

Mehedi Hassan Nayeem  
17101261

*Misbah Uddin Sagor*

---

Misbah Uddin Sasgor  
17101283

# Approval

The thesis/project titled “Dynamic Spam Detection System and Most Relevant Features Identification Using Random Weight Network” submitted by

1. Syed Mahbubuz Zaman (16201017)
2. A.B.M Abrar Haque (17101078)
3. Mehedi Hassan Nayeem (17101261)
4. Misbah Uddin Sagor (17101283)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 08, 2021.

## Examining Committee:

Supervisor:  
(Member)



---

Moin Mostakim  
Senior Lecturer  
Department of Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)



---

Dr. Md. Golam Rabiul Alam  
Associate Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

Prof. Mahbub Majumdar  
Chairman  
Department of Computer Science and Engineering  
Brac University

## Abstract

Nowadays e-mail is being used by millions of people as an effective form of formal or informal communication over the Internet and with this high-speed form of communication there comes a more effective form of threat known as spam. Spam e-mail is often called junk e-mails which are unsolicited and sent in bulk. By these unsolicited emails, the Internet users are hugely impacted in terms of security concerns as well as being exposed to contents that are not appropriate for certain users. There is no way to stop spammers using static filters because almost every other day they find a new way to bypass the filter. New techniques are introduced to elude this system. In this paper, a smart and dynamic(adaptive) system is proposed that will be using Random Weight Network (RWN) to approach spam in a different way and meanwhile this will also detect the most relevant features that will help to design the spam filter. A spam filter with the capability of identifying spam automatically will also be embedded in the proposed system. Also a comparison of different parameters for different RWN models have been shown to determine which model works best with what parameters under different situations.

**Keywords:** Spam filtering; Email spam detection; Feature analysis; Long Short Term Memory; Bidirectional Long Short Term Memory; Gated Recurrent Unit; Evolutionary; Random Weight Network; Feature selection; Natural Language Processing

## **Acknowledgement**

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our supervisor Moin Mostakim sir for his kind support and advice in our work. He helped us whenever we needed help.

In this challenging time of COVID-19 pandemic, we faced many challenges but with the blessings of almighty Allah and our parents and helpful teachers we have overcome them.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgment</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>Nomenclature</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction	1
1.2 Problem statement	1
1.3 Types of spam	2
1.3.1 Comment Spam	2
1.3.2 Trackback Spam	2
1.3.3 Negative SEO Attack:	3
1.3.4 E-mail Spam	3
1.4 Life cycle of spam	3
1.4.1 Spammer	3
1.4.2 ISP	3
1.4.3 User	4
1.5 Research objective	4
<b>2 Literature review</b>	<b>5</b>
2.1 Related Work	5
2.2 Popular Spam Handling Techniques	8
2.2.1 Yahoo Mail Spam Filtering	8
2.2.2 Gmail Spam Filtering	8
2.2.3 Outlook Spam Filtering	9
2.2.4 Content-Based Filtering Technique	9
2.2.5 Case Base Spam Filtering Method	9
2.2.6 Heuristic or Rule-Based Spam Filtering Technique	10
2.2.7 Memory Based Spam Filtering Technique	10
2.2.8 Adaptive Spam Filtering Technique	10
2.3 Modern Techniques	10
2.3.1 DKIM- Domain Keys Identified Mail	10

2.3.2	SPF- Sender policy Framework . . . . .	11
2.3.3	DMARC- Domain-based Message Authentication, Reporting and Conformance . . . . .	12
<b>3</b>	<b>Methodology</b>	<b>13</b>
3.1	Neural network . . . . .	13
3.2	Recurrent Neural Network . . . . .	14
3.3	Vector to Sequence Model . . . . .	14
3.4	Sequence to Vector Model . . . . .	14
3.5	Sequence to Sequence Model . . . . .	15
3.6	Encoder-Decoder Architecture . . . . .	15
3.7	Problems Related to Diminishing Gradients . . . . .	16
3.8	Effects of Diminishing Gradients . . . . .	17
3.9	Solution to Vanishing gradients . . . . .	17
3.9.1	Skip Connection . . . . .	17
3.9.2	Remove connection length . . . . .	18
3.9.3	Leaky recurrent Units . . . . .	18
3.9.4	Gated Recurrent Networks . . . . .	18
<b>4</b>	<b>Data and Model Workflow</b>	<b>20</b>
4.1	Dataset . . . . .	20
4.2	Data Preprocessing . . . . .	21
4.3	Tokenization of the Cleaned Data . . . . .	21
4.4	Sequencing . . . . .	21
4.5	Model Selection . . . . .	22
4.5.1	LSTM . . . . .	22
4.5.2	GRU (Gate Recurrent Unit) . . . . .	23
4.5.3	Bi-LSTM . . . . .	23
4.6	Implementation . . . . .	24
4.7	Workflow Diagram . . . . .	25
<b>5</b>	<b>Result Discussion</b>	<b>26</b>
5.1	Metrics of Evaluation . . . . .	26
5.2	Model Accuracy and Confusion Matrix . . . . .	27
5.2.1	Accuracy and Confusion Matrix LSTM . . . . .	27
5.2.2	Accuracy and Confusion Matrix BiLSTM . . . . .	29
5.2.3	Accuracy and Confusion Matrix GRU . . . . .	30
5.3	Output Analysis . . . . .	31
<b>6</b>	<b>Conclusion and Future Work</b>	<b>33</b>
6.1	Future Work . . . . .	33
6.2	Conclusion . . . . .	34
	<b>Bibliography</b>	<b>37</b>

# List of Figures

2.1	Workflow of DKIM . . . . .	11
2.2	Workflow of SPF . . . . .	11
2.3	Workflow of DMARC . . . . .	12
3.1	Vector to Sequence Diagram . . . . .	14
3.2	Sequence to Vector Diagram . . . . .	15
3.3	Sequence to Sequence Diagram . . . . .	15
3.4	Encoder-Decoder Architecture . . . . .	16
3.5	Simple RNN . . . . .	16
3.6	When $W > 1$ , eigen vectors explode to infinity. When $W < 1$ , eigen vectors vanish . . . . .	17
3.7	Skip Connection . . . . .	17
3.8	Remove connection length . . . . .	18
3.9	Leaky recurrent Units . . . . .	18
3.10	Gated Recurrent Networks . . . . .	18
4.1	Enron1 parsed to CSV . . . . .	20
4.2	LSTM . . . . .	22
4.3	Bi-LSTM . . . . .	23
4.4	Proposed Neural Network Model . . . . .	24
4.5	Workflow Diagram . . . . .	25
5.1	Model Accuracy and Confusion Matrix using Adam . . . . .	27
5.2	Model Accuracy and Confusion Matrix using Nadam . . . . .	27
5.3	Model Accuracy and Confusion Matrix using RMSProp . . . . .	28
5.4	Model Accuracy and Confusion Matrix using Adam . . . . .	29
5.5	Model Accuracy and Confusion Matrix using Nadam . . . . .	29
5.6	Model Accuracy and Confusion Matrix using RMSProp . . . . .	29
5.7	Model Accuracy and Confusion Matrix using Adam . . . . .	30
5.8	Model Accuracy and Confusion Matrix using Nadam . . . . .	30
5.9	Model Accuracy and Confusion Matrix using RMSProp . . . . .	30
5.10	Output Analysis . . . . .	31
6.1	Proposed Future Model . . . . .	33



# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

*ANN* Artificial Neural Network

*BiLSTM* Bidirectional Long-Short Term Memory

*CNN* Convolutional Neural Network

*GRU* Gated Recurrent Unit

*LSTM* Long-Short Term Memory

*NB* Naive Bayes

*RNN* Recurrent Neural Network

*RWN* Random Weighted Network

*SVM* Support Vector Machine

$\sigma$  Sigmoid

# Chapter 1

## Introduction

### 1.1 Introduction

E-mail (electronic mail) is the most popular manner of communicating records and facts among individuals in an organization or group. The cheap and easy way of transferring records is once in a while exploited as the receiver is ignorant of the unsolicited mails. Market rivals' always try to target company executives who have access to confidential information with ill-natured emails to get those data which can cause a huge loss to the company. Most of the previous researches and researches done in recent times to detect spam emails are very much restricted to port addressing protocols. Even most commercial strategies to combat spam are limited to static filters and checking manually.

### 1.2 Problem statement

Spam, which is responsible for flooding user mailboxes. It wastes users' time by reading them, and prices billions of dollars in wasted information measure and disk space for storing [10]. This huge number of spam e-mails are being thrown over the Internet has damaging effects on the storage units of e-mail servers, communication information measure, central processor power and user time [24]. Some spams could contain malware that steals user data, like Mastercard info., social insurance variety and e-mail passwords. Thus, spam filtering is critical for user safety. Spam is responsible for around 14.5 Billion emails a day all over the world which is more than 45% of the total number. The most typical approaches are using filters that display messages based on the existence and persistence of known words and common phrases which is basically a pattern recognizer and another one is to simply creating a blacklist(automatically rejects emails from known spammers) and a whitelist(automatically accepts emails from trusted correspondents).The most crucial drawback within the these two techniques is that it depends on the spammers' self-satisfaction by forwarding that they will not change their identities and or change the pattern of their mails by using differently designed vocabularies. They were unable to eliminate the risks of the likelihood that the recipient will miss a real legitimate e-mail from a known or expected correspondent with an unfamiliar email address, like correspondence from an old friend, or an invoice of something that has been sent via email. An in-depth rationalization of those techniques is given in [9]. A solely practical approach to managing numerous bulk spams is to

use automatic machine learning algorithms that will notify the randomly changing spam attributes. Spammer strives their best to beat the spam filters, which unintentionally enables the e-mails to become a lot more conspicuous. A solely practical approach to managing numerous bulk spams is to use automatic machine learning algorithms that will notify the randomly changing spam attributes. Spammer strives their best to beat the spam filters, which unintentionally enables the e-mails to become a lot more conspicuous. In spam classification certain ensemble strategies such as AdaBoost exploitation MLP, Single MLP, Reinforcement Learning, Naïve Bayes and Mixture of Experts (MOE) have shown positive results in [8]. Authors of paper [14] explored; however, the classification performance is often suffering from the dataset scale. Accuracies of 92.7%, 97.2%, and 95.8% were achieved for a dataset of size a thousand by SVM, NB, and J48 severally. Training and testing a dataset of 1000 instances SVM, NB and J48 has achieved accuracy of 92.7%, 97.2% and 95.8% accordingly. Previous ANN and BP-based deep learning methods have certain problems that have been solved by the Random Weight Neural Network(NNRW). NNRW selects hidden weights and biases randomly within a given range and keeps on doing so throughout the whole training process but in the hidden layer and in the output layer the weights and biases are selected with utmost sincerity using different analytics. NNRW has a much faster training rate with fairly acceptable accuracy compared to those traditional BP based training strategies. NNRW is easy to implement and its universal potential for approximation has been observed in theory. [1][11].

## **1.3 Types of spam**

### **1.3.1 Comment Spam**

Comment spam can be done by two ways and one of them is user generated where user comments or post spammy links on websites and the other one is through scripts where bots used to put links to the website and this is usually hit by the automated script which automatically comment on someone's website comment section. Overall ranking of anyone's website can be affected if anyone having the comment spam on one part of their website that is owned only on the comment section. Comment spamming also puts a negative impact on the user's experience. So, if the users experience will be negative then there will be different matrices which are associated with users experience and for this site will also get affected. DDoS and Bots attack are also included in the spam comment.

### **1.3.2 Trackback Spam**

Trackbacks were created to be a useful tool where web admins are being notified manually by copying other websites' trackback hyperlinks and then using this as a way of making connections to the webmasters. Also, newly opened websites can easily gain popularity by using trackback which actually creates links to famous or popular content.

### **1.3.3 Negative SEO Attack:**

Negative SEO attack is used for ranking up someone's website and also used to downgrade the competitors ranking. Spam SEO is a negative SEO and it basically created from website specially automated spam websites and this happens without being aware of the victim it created toxic links to websites and as a result it harms victims personal profile by increasing the links risk.

### **1.3.4 E-mail Spam**

Email spam also known as junk emails and more of them are unwanted, malware emails which are sent out for very small kinds of interaction. That means email has lots of links, images and that links, images redirected to some webpages and this redirection will help spammers to get details like bank details or identity so that they can spam using someone's identity. Basically, spammers collect email addresses from customer lists, newsgroups, websites or chat rooms and viruses which harvest users address books and are sold to other spammers. Different companies use different spam filtering so that they can detect the spam email and google is one of them for which users are not attacked by spammers easily. But there has also a downside which is sometimes valid emails are getting into spam folders for which users got affected.

Among these all spam, we are focusing mainly on the Email spam, and we are trying to develop a program which can easily detect which emails are spam and which mails are not. Whenever the program detects any spam in an email, it will notify the user.

## **1.4 Life cycle of spam**

Spammers are now using much more intelligent methods to make their spam messages more reliable for the search engines. They can smoothly go to the top of the search engines by suggesting multiple keywords. To avoid being attacked by them, we need to know how things work. In this section, we are going to talk about the stakeholders of spam. The stakeholders of email spam include:

### **1.4.1 Spammer**

Spammer starts this life cycle by creating spam emails to gain access to other accounts, to get revenue, to rank up in search engines, and promote products. They use so many techniques so that their spam cannot be detected in general spam detectors used by the ISP or the search engines. In the meantime, all of this motivates a spammer to be more creative and make spam more realistic.

### **1.4.2 ISP**

ISPs play as the media between spammers and the users. Both stakeholders are useless if there is no Internet Service Provider (ISP). Due to the restricting spammer to continue spamming ISPs would have to maximize their bandwidth for the users, which indirectly increases their operational cost.

### 1.4.3 User

The crucial part and also the victim of all of these. The whole cycle depends on the users' choice whether to open the spam email or not. If he opens it, the cycle continues. Otherwise, it ends there. Unintentionally, the human mind gets excited when they see something different or exciting. So, when a spammer gets more creative and attaches all the exciting sites with his spam emails, then users open those, and this cycle starts. Sometimes, users believe in anti-spam programs blindly.

## 1.5 Research objective

For many reasons, we have seen the manual filtering method fail in the past, and for many more upcoming reasons, we can see this contemporary pattern-based filtering not working for a long time. We need to develop something that can learn from its past and evolve itself according to the necessity of that very situation. As technology has evolved revolutionarily and now, we can provide our machines with intelligence, they can do much more than before and why not take advantage of this function. In the past and even now learning-based classification techniques have been used to develop spam filters which have proven to be very efficient to a certain extent and can save much time but most of the studies have focused on the accuracy level and how can they make it more accurate and hardly focused on the features that can be extracted and the predictive skills of these algorithms. Now our goal is to create an email spam detection system where the Feature Selection(FS) will be automated and the classification wholly depends on the two major stages. The selection stage works as a wrapper approach, while the classification is handled by RWN. The proposed system's main advantages include achieving high accuracy for classification, and measuring the impact of the parameters while training. Secondly, this system will help to recognize which feature would be more significant in the detection stage. Therefore, let's sum up the main contribution of our system in the following points:

- The hybrid-RWN is established as a new effective spam identification model.
- The suggested model determines the most important characteristics in the detection process automatically.
- Unlike other wrapper-based FS methods, Auto RWN employs random weights to gain from its possible improved generalizing capacity. The Auto RWN uses randomized weight as a simple classifier.
- Determine spam of email through using different types of Neural Network classifier.
- Collect datasets from different kinds of websites like Kaggle.
- Predict which classifier will be fit for our selected dataset.
- Predict the percentage of spam of email from the selected classifier; for example, we use LSTM, Bi-LSTM, GRU.
- Create a standard model for getting better accuracy.

# Chapter 2

## Literature review

### 2.1 Related Work

Arram et al. [19] proposed a hybrid combination of Artificial Neural Network (ANN) and Genetic Algorithm (GA) to filter spams. ANN and GA are getting popular among researchers. This proposed structure aims to enhance the accuracy of spam detection by classification of email content using this hybrid of ANN and GA. In the learning phase, this technique was applied on almost 60% of the full data sets and the other 40% was used for testing purposes. Back Propagation (BP) algorithm is used in this work. Besides, GA has been used to determine the parameters to enhance BP learning to be more accurate. Among so many criteria – False positive (FP) and False negative (FN) is considered to measure the test's performance. Using the GA optimization method, the process can decrease the FN and FA, which increases the accuracy of the results. Also, the result of this technique is about 93.71%.

Most of the spam filtering techniques are predicted on text categorization methods. Thus, filtering spam activates a classification problem. Christina et al. [16] used a filter to extract feature vector from email within which rules are framed. Since discrimination characteristics do not seem to be well defined, it is more convenient to use machine learning techniques. The three machine learning algorithms, C 4.5 Decision tree classifier, the multilayer perceptron and Naïve Bayes classifier, are used to learn the classification models. The training dataset, spam and legit message corpus are generated from the emails that we received from our institute mail server for six months. The mails are analyzed, and 23 rules are identified that significantly ease classifying the spam message. The corpus consists of 750 spam messages and 750 legitimate messages. The feature vectors are extracted from the corpus by analyzing message header, keyword checking, whitelist/blacklist. They generated spam and legit message corpus from the newest mails and employed machine learning techniques to create our work model. The model is evaluated upon 10-fold cross-validation and observed that Multilayer Perceptron classifier outperforms other classifiers and therefore, the false positive rate also shallow compared to other algorithms.

Another famous intelligent detection system of the spam messages is introduced by Faris et al. [26] which mainly is based on the Genetic Algorithm (GA) and Random Weight Network (RWN) with an extra automatic identification capability to

extract the most relevant features of from the spam emails during the process. They proposed that this system comprises five stages: Feature extraction, Feature Selection (FS), model development, evaluation and assessment, and feature importance analysis. Moreover, methods like the conventional Backpropagation (BP) needs its parameters of all kind to be set up manually where, on the other hand, RWN does not need any human intervention in its process. The research also mentioned that this RWN selects the weights and the hidden biases randomly, then it determines the output weights analytically using Moore-Penrose generalized inverse [4]. Besides, they used three public spam corpora for constructing their datasets which are SpamAssassin, LingSpam, and CSDM2010 Corpus. To eliminate irrelevant features, they applied FS, mainly a wrapper-based approach, which also helps to decrease the complexity of vector space in data and raise the classification accuracy up. As we know, wrappers are preferable than other methods such as filters when accuracy is the top priority than the speed of the task. Altogether, the proposed Auto-GA-RWN algorithm can be described in five main steps which are Initialization, Genotype-Phenotype mapping (FS and RWN construction), Fitness evaluation, Selection and reproduction, and Termination. They also showed some statistical difference between some similar procedures to be more precise about their technique.

Another research by Hu et al. [15] shows that Complex-Valued Neural Network (CVNN) can be useful to detect spam. This work consists of these sides: **1.** They proposed a model based on the CVNN to classify emails, and also it changes the input of the email to the 2-dimensional vector data stream. **2.** Our system extracts the standard and essential features from the received emails, which are also called Behaviour-based Characteristics. **3.** CNN's input layer includes three layers, 30 neurons, one hidden layer with 26 neurons and two neurons as the output layer.

They also collected 3000 samples from the HUST mail server. Besides, about 30 high-frequency sensitive words are being extracted from those samples to create the vectors. Moreover, the results show that CVNN converged more stable than BP. Finally, this system can predict spams to 98%, which is much better than BP.

In this paper, Zhang et al. [23] came up with a spam detection method which focused on decreasing the False Positive error of non-spams which are mislabeled as spams. For this, they used wrapper-based feature selection methods to extract essential features, the decision tree as the classifier model and C4.5 as the training algorithm, and cost matrix to assign different weights to the two errors – FP & FN. The Binary Particle Swarm Optimization (BPSO) with the mutation operator (MBPSO) was used as the subset search strategy for the method. They used a dataset of 6000 emails to test and train this method. The accuracy of the decision tree with FS by MBPSO was 94.71%. Results showed that MBPSO performs so well than many other conventional methods using FS.

Shahane et al. [22] proposed a technique for implementing ANN on an essential 8-bit processor where learning is run on the same stage. They thought if a device can adapt its way of functioning as its user, devices can be useful and flexible for work. In order to do so, ANN is the way out. Basically, this paper focuses on an embedded system consisting of ANN and learning algorithms. First of all, they used a

database of possible inputs with their related outputs given to a training algorithm. That algorithm produced the weights of each node. Furthermore, in this process, to get the minimum error, the number of epochs must be kept different from each other, and it takes less time than others.

This article by Katasev et al. [25] states a solution for email messages classification with the help of Neural Technology. This system mainly analyzes basic spam filtering methods; spam mails repeat detection and Bayesian filtering according to words. Firstly, the neural network needs to be trained for this job. According to them, the studies show that the developed ANN model is adequate to classify spam and non-spam messages. Thus, this paper introduced the possibility of the effective neural network model use for this classification.

Multilayer Perceptron (MLP) is one of the most effective spam detection techniques. As we all know nowadays, spam detection is so crucial to be successful in life. So, Leng et al. [20] in their paper came up with an idea to improve MLP web spam detection accuracy to its limit. MLP neural network is mainly known for the flexible structure and non-linearity transformation to cope up with the latest spam patterns, and that is why, in this paper, they broadly explained about it. Spams – link-based or content-based both can be tested in the MLP network. They used WEBSpAM-UK2006 AND WEBSpAM-UK2007 datasets for evaluating the performance of the proposed classifier. Then the whole performance was compared with SVM, which is enormously used for spam detection. These experiments also outperformed the SVM by 14.02% on the former dataset and up to 3.53% on the later dataset.

Aside from SVM and DT, neural networks have emerged as a vital classification tool and have been demonstrated to be a competitive alternative to traditional classifiers [4]. There are few researchers [13], [18] who used neural networks for Webspam classification. However, they did not mention the architecture, which is crucial for the neural networks' performance. Furthermore, even though the latter authors [18] have shown that LAD Tree [6] outperforms both SVM and neural networks, there are no clear explanations on the supervised learning algorithms for their experiments on machine learning models. Closest to our paper is Noi et al. [17] which use probability mapping-graph self-organizing maps for clustering, and then graph neural networks for classifying. However, the training time for a mixture of unsupervised and supervised is computationally expensive.

This paper [21] shows the improvement of the Artificial Neural Network (ANN) by using the Memetic Algorithm (MA) which also evaluates its activity on the UCI spam base dataset. The MA algorithm incorporates the local search capacity of Simulated Annealing (SA) and the global search capability of a Genetic Algorithm (GA) to optimize ANN parameters. In this paper, they faced many differences in parameters, mechanisms, and architectures used to optimize the network's performance. This paper also shows us how to gain stability between MA and GA. Furthermore, the dataset used in this experiment consists of 4601 instances and also the class distribution comprised of 1813 spam and 2788 ham or non-spam messages. They split the instances in 80:20 ratio for training and testing, and also, they were randomly selected. After this training, the results were tremendous and hence con-



firmed Rosin et al. [3] that the mutation rate in MA can be more adventurous in its role. At last, they proved that MA effectively fights the genetic drift and guarantees better convergence and adequate performance with lesser training epochs.

## 2.2 Popular Spam Handling Techniques

Spam is a problem that agitates even the biggest ISP's. To combat this situation effectively email providers are leaning more towards machine learning techniques and AI algorithms. Instead of using protocol checking methods popular in the past the implementation of AI and ML techniques ensures that the filtering is done more correctly. The AI can learn to recognize spam from the pre-existing datasets and huge archives. It can also learn the spam detection patterns intuitively like a human brain and predict spam attacks that might take place in the future. It can also dynamically adjust to the random variables to give accurate results. Google is confident enough to claim that its AI Deepmind can sort about 99.9% of the spam messages. The only back draw towards this approach is that about one in one thousand messages manages to pass through such secure measures. Some of the critical features implemented are safe browsing through https websites and restricting browser access to questionable websites. Google's own estimate is that about one third to half of the mails in Gmail are spam. To prevent phishing and spam google implements image selection or are you a robot? validation process. Which may seem counterintuitive at first because it delays the user's ability to send email. On the contrary, it's quite an effective measure as it helps to validate spam emails thus ensuring user safety of resources and time and in the long run. Another approach is to delay mailing intentionally of some message to put them under supervision. The process of filtering through segregation is easier because it's easier to dissect the mails individually. This purposeful delay influences just about 0.05% of messages because the main algorithms can inference from the thorough inspection. Already there are a lot of spam filtering processes that include filtering based on trustworthiness of the email sender. Few of them are described here to create a contrast between the existing and proposed model and furthermore, highlight the flaws in these systems.

### 2.2.1 Yahoo Mail Spam Filtering

Yahoo mail is vastly known within the world as the first free webmail provider with more than 320 million users. Yahoo is being successful in detecting the spam emails over the year with the help of their some basic ways which consists of email content, URL filtering and spam complaints by users. They focus on filtering spam by domains rather than filtering by IP addresses. Mainly, the techniques of filtering used by Yahoo mail includes a special method for preventing any legitimate user to be recognized as a spammer.

### 2.2.2 Gmail Spam Filtering

Google's data center follows some rules to find out emails authentication and tells the user which one is spam or not. The following principle by google leads to the

highlight of spam emails. Firstly, Gmail spam filters look at the sender's email address and find out whether the email address is in the blacklist or not. If the email has been listed in the blacklist section then the spam filter will not allow the email to send any kind of messages. Moreover, Gmail uses a huge database for hypocritical or malicious links where the filter is detecting the words and phrases which are suspicious like "How to get rich in one day", "How to earn money" etc. But spammers also know how the Gmail filter works and they are finding out new ways continuously like using different characters. To overcome this Google is using AI and also their built-in machine learning function. AI calculations created to join and rank huge arrangements of Google indexed lists permit Gmail to interface many variables to improve their spam order. Spam sifting mainly chips away at the establishment of filters settings that are constantly refreshed with the development of best-in-class devices and calculations.

### **2.2.3 Outlook Spam Filtering**

The previous name of Outlook was Hotmail and Microsoft changed it recently. Microsoft basically follows some rules to filtering the spam emails which are sender reputation data, sending reputation, complaints they get from users of other users, engagements of users, authentication and blacklists. First of all, Microsoft spam filtering is heavily based on sender reputation data or SRD. There is a separate SRD panel in Microsoft panel and they send invitations to the trusted user to be the members of the sender reputation panel. Users can vote emails based on what they have received in their inbox whether it is spam or not and their votes valued so much and also these votes carry lots of weight to filtering the spam message. Moreover, they change their panel frequently so that they can get more information and votes from other users. Sending reputation is also one of the main factors which has been used for filtering spam. Sending reputation measures based on how others engage with his/her email, how frequently he/she sends email to unknown users and also how others give complaints against him/her. Microsoft also depends on engagement of the user like how recipients open his/her email and how they reply to his/her email and whether they keep that person's email in the junk folder or not and also whether the recipients delete that person's email without reading it or not. Microsoft also follows DKIM, DMARC and SPF methods to verify the mail is coming from the intended domain or not. If not then they send that email into the junk email or spam email section.

### **2.2.4 Content-Based Filtering Technique**

Using simple machine learning classifiers such as SVM, Naïve Bayes, K-NN it is possible to filter spams automatically. These methods usually use linear regression and similar distribution techniques to access word count, phrases and repetition to determine the emails fate as spam or ham. [16].

### **2.2.5 Case Base Spam Filtering Method**

Using a collection model, the spam and ham emails are gathered from the all of users' mail inbox. This is a common technique that uses predetermined processing which

includes feature extraction, segmentation and selection of mails for tokenization and padding process within the client's domain. The dataset is split into train test format so that it can be put through machine learning algorithms. This in return will isolate the malicious mails from the safe mail. [16].

### **2.2.6 Heuristic or Rule-Based Spam Filtering Technique**

SpamAssassin is a popular spam corpus. Many companies avoid spam just by following common heuristics while sending out or receiving mails. The Mailing format is predetermined by the companies by using reg-ex. The mails that are tagged as spams are used as reference for future mails. Every time someone receives a mail, a score is assigned to the mail signifying its probability of being spam. The ranking method is determined by mail admin. So, sometimes the rules become outdated or overruled by the attackers very easily.[16]. A fantastic illustration of a standard based spam channel is SpamAssassin [12].

### **2.2.7 Memory Based Spam Filtering Technique**

AI algorithms are used to member prior knowledge to categorize the incoming emails into subsections to justify whether or not it is suspicious or safe. If an email shows signs of being a spam the AI models separates it from the inbox. Mainly it works by ascribing the input texts into vectorized values that run through algorithms to give out predictions. A popular clustering algorithm is K-NN (k nearest neighbors). KNN works by clustering the dataset into chunks or clusters that determine the prediction of the new value [5].

### **2.2.8 Adaptive Spam Filtering Technique**

The mail corpus is divided into different strata. Each level signifies the level of confidence based on the email receiver. On the basis of the comparison measure system, it is decided if the mail is to be sent to inbox or quarantined. Thus, the method utilizes mail swarming techniques to filter spam [7].

## **2.3 Modern Techniques**

Some of the proven methods for detecting spams are DNS based techniques mainly known as DKIM, SPF, DMARC.

### **2.3.1 DKIM- Domain Keys Identified Mail**

DKIM can be considered as a standard form of email verification. It validates the authenticity of emails by registering a public key. It uses cryptography to encrypt the sent emails and performs cross check with the received emails. This process ensures that the emails have not been tampered with along the delivery path. This protocol works over the SMTP as an additional safety measure. The domain administrator ensures that a text record is present for the specified domain that works alongside with the system DNS. DKIM works by assigning an unique signature to

the message header whenever a message is sent by an outsider. When an user inside the domain receives emails, the DNS server looks up the public DKIM signature for that domain. That key is used to decrypt the signature and then it cross validates with the existing key computed by the DKIM. When the validation is completed the mail reaches the user and the domain administrator acknowledges the safety of the mail.

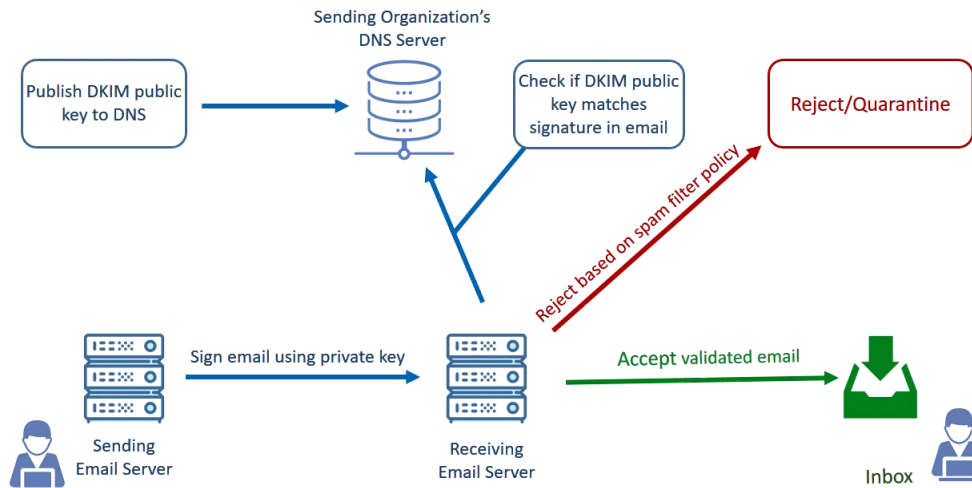


Figure 2.1: Workflow of DKIM

### 2.3.2 SPF- Sender policy Framework

The domain administrator keeps the record of the domain in text format known as SPF records. This format uses the email sender's Ip addresses and mail servers. When the delivery process is begun the receiving mail server searches the SPF record related to the senders domain which may include the Ip address. It checks whether or not the sender is a member of the trusted group. If any malicious mail is sent from an unknown sender, it is immediately detected by checking the SPF record. The domain administrator can take action against this by isolating the mail in quarantine, rejecting the mail altogether or simply letting it pass.

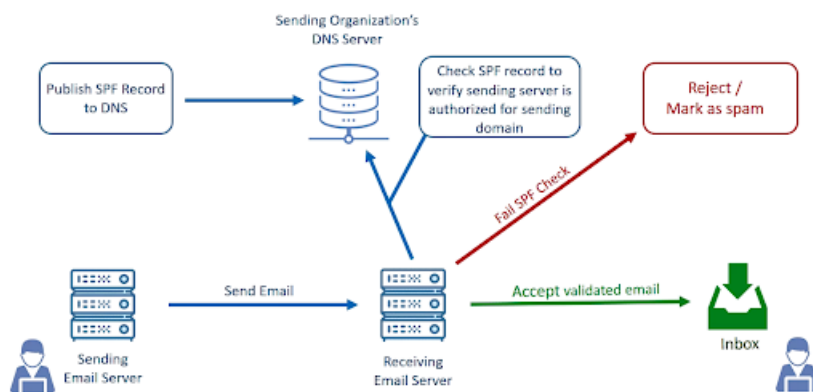


Figure 2.2: Workflow of SPF

### 2.3.3 DMARC- Domain-based Message Authentication, Reporting and Conformance

The full functionality of DKIM and SPF is implemented by the use of DMARC, which is not an email authentication system in itself but works hand in hand with DKIM and SPF. Within the DNS the Domain administrator publishes a company policy that makes suggestions to the mail hosts and tells them what to do when an invalid email has failed validation process through not being included in the authorized list or for providing a broken DKIM. Some mails claiming to be forms are also reported to the domain administrator by DMARC and also necessary steps are recorded. The DMARC can be configured in a way that the receiver will not get any mails that have failed either SPF or DKIM or both. Also there is a measure to send detailed reports to the domain administrator.

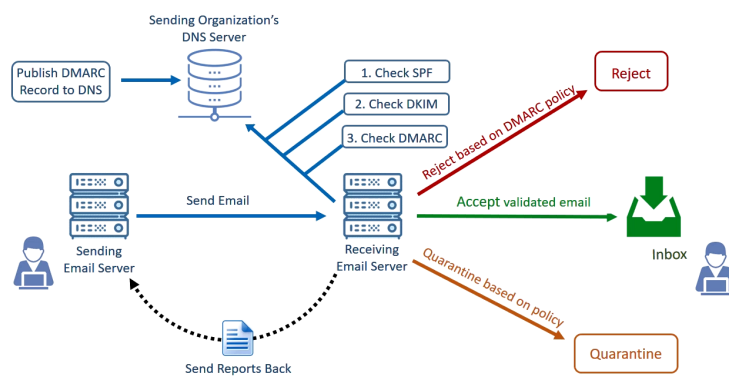


Figure 2.3: Workflow of DMARC

# Chapter 3

## Methodology

### 3.1 Neural network

Neural networks are modelled after the brain cell or neurons. Each neuron receives stimulus from the synapse of the previous neuron. This way it can articulate involuntary and voluntary muscles. The neural network works by activating each perceptron. Each perceptron is a function that multiplies a weight to the axis or vectors and adds bias to it. This is then passed onto the next perceptron. An activation function is used to generate the output. This way a simple feed forward neural network makes predictions based on the existing data. In order to dive into deep learning. The neural networks need to develop both in concept and complexity. Through the introduction of back propagation and gradient descent, a neural network is able to learn actively. Neural networks can take both approaches while learning from a dataset.

- **Supervised Learning:** In this learning the operator sets the parameter for the network and at the same time selects the optimizers, learning rate and loss function. The data set is pre processed and the algorithms give out inputs accordingly.
- **Unsupervised Learning:** In unsupervised learning the neural network works on its own to figure out the meaning of a dataset without any user inputs. Although it may seem that the pattern of learning is unconventional. Sometimes a deep insight can be generated while executing this method. feed forward

$$NN = \sigma(w^1x^1 + w^2x^2 + w^3x^3 + \dots w^nx^n + bias)$$

A neural network can be illustrated in terms of mathematical function of differentiability that takes in a value of X and multiplies it with a weight and adds a bias to it. To put it simply every perceptron takes one function and outputs another function. This neural network is capable of mapping one vector function into another vector function or scalar function depending if the problem is classification or regression respectively.

## 3.2 Recurrent Neural Network

Recurrent Neural Networks is an extension of Feedforward neural network with the addition of sequences ending up with numerous architectures. These architectures can be used in various applications. RNN activation functions can flow in a loop due to the existence of one backfeed connectivity. This in turn helps to predict temporal prediction and sequence recognition. MLP is one of the most common architectures in RNN. MLP has a type of memory due to its multidimensional vector mapping features. the presence of stochastic activation functions along with interconnected neurons with uniform spread. Through the gradient descent process the back propagation networks learn to reduce the loss. Recurrent neural networks can be of multiple types like vectored sequence models, sequence to vector models and sequence to sequence models.

## 3.3 Vector to Sequence Model

Vector to sequence model architecture is one of the most popular sequencing models. Here the desired length sequence is generated based on the length of the vector. Image captioning is one of the innovative applications of this model. Images can be put into the model to create outputs that are text or audio-based description of the input image.

### Vector to Sequence

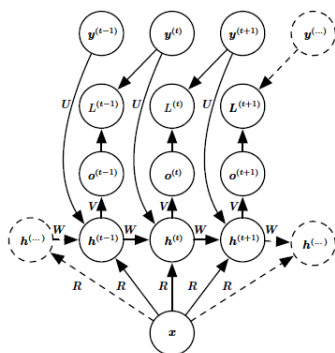


Figure 3.1: Vector to Sequence Diagram

## 3.4 Sequence to Vector Model

A second architecture that we discussed is the sequence to vector models. For instance, we take inputs as words in a sequence and outputs can be generated in vector form which is machine readable. Let's say for example the inputs are the words of a movie or product review can be inputs to the model to generate an analysis of sentiment. Here the output would be a vectorized representation of the inputs which can clearly label the review or sentiments as positive or negative.

Sequence Input, Single Output

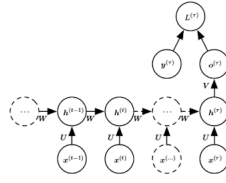


Figure 3.2: Sequence to Vector Diagram

### 3.5 Sequence to Sequence Model

The third architecture we looked at is the sequence-to-sequence models. Here the input is a sequence which is mapped into the outputs which are also sequences. With sufficient training, the outputs will be able to predict the next words which are provided in the input of a sequence of words. At some point it can create sentences on its own once it has built a well enough dictionary or word-level language model.

#### Recurrent Hidden Units

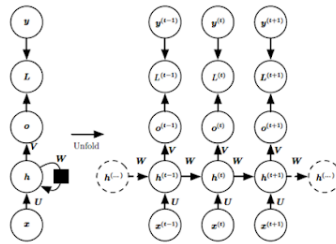


Figure 3.3: Sequence to Sequence Diagram

### 3.6 Encoder-Decoder Architecture

In real life scenario we don't find applications of models with equal number of inputs and outputs. In case of translation from English to Bengali, the output sequence should not be word to word translation rather an interpretation of the input sequence. That is also true in case of summarization of an article which requires the output to be shorter than the input. In order to resolve the problem, we need a newer model that is capable of changing the length of inputs and outputs relevant to the content. This type of architecture is named the encoder/decoder architecture. Firstly, the encoder that converts the sequence to a vector. Secondly, in case of English to Bengali translation those vector inputs are turned into a new set of sequences. Sentences are converted into sequences this way and the words are kept in the  $x(t)$  neurons. Through the  $y(t)$  neurons, Bangla words are kept in the encoding part. Meaning vector is formed right after taking in a sentence in the form of a sequence. The decoder receives this meaning vector further translates it to a sequence which is a Bengali sentence



## Sequence to Sequence Architecture

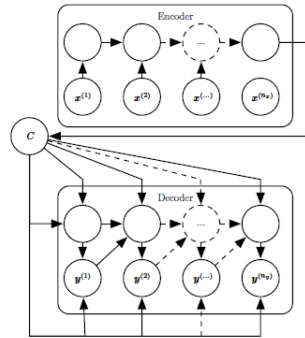


Figure 3.4: Encoder-Decoder Architecture

### 3.7 Problems Related to Diminishing Gradients

Theoretically, these sequences can be infinite. There's a problem. Let consider the case of a recurrence on some scalar  $x_0$ , implemented in a RNN that consists of no hidden unit. After  $n$  time the value of input changes into  $x_n$  because that system is a discrete dynamical system. In this network, the scalar way  $W$  can be achieved by backpropagation through time (BPTT) algorithm. What will be the changes for a

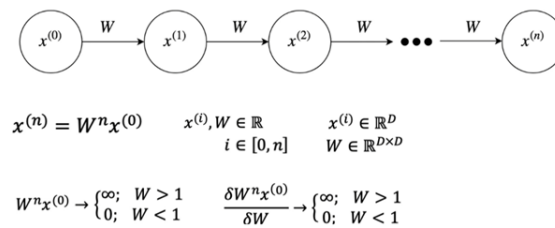


Figure 3.5: Simple RNN

higher value of  $x_n$  for a minimal value of  $n$ . Considering the case where  $W > 1$ , then  $W^n x(0)$  explodes, and  $W$  is somewhat less than 1 then  $W^n x(0)$  would go closer to zero. In case of forward propagating neural networks this diminishing value would carry over to the next nodes as a result gradient will also diminish. This phenomenon can be generalized into a vectorized form and saved as a matrix where  $X^T$  would be a vector. Furthermore, the transformed matrix would be  $W$ . Considering,  $W > 1$  the entry values would be assigned eigenvectors in the transform matrix.  $W^n$  go towards infinity and eventually explode. Which ultimately suggests that the values will be too high for matrix manipulations and eventually the information will be lost.

The effect is quite the contrary when considering  $W < 1$ , the entry values after being assigned to eigenvectors goes closer to 0 and as a result the eigenvectors lose their meaning as it reaches zero. This in turn would lose input information because of the vanishing values.

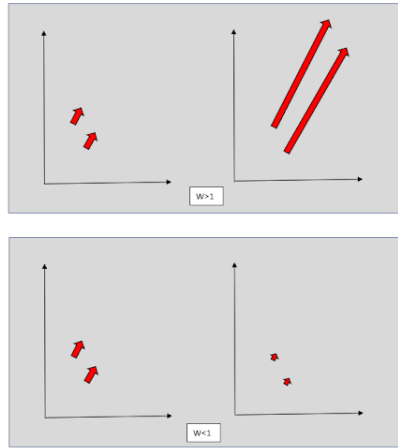


Figure 3.6: When  $W > 1$ , eigen vectors explode to infinity. When  $W < 1$ , eigen vectors vanish

### 3.8 Effects of Diminishing Gradients

The effect of vanishing and exploding gradients is much worse in RNN than it is for traditional deep neural networks. This is because DNN's have different weighted matrices between layers so if the weights between the first two layers are greater than 1 then the next layer can have matrix weights which are less than 1 and so their effects would cancel each other out. In the case of RNN ends the same weight parameter recurs between different recurrent units so it's more of a problem because it cannot be cancelled out.

### 3.9 Solution to Vanishing gradients

#### 3.9.1 Skip Connection

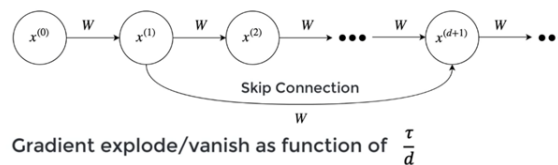


Figure 3.7: Skip Connection

Hochreiter [2] in 1991 found a way of dealing with this problem of vanishing and exploding gradients, the first approach we can take is to apply skip connections. Applying additional edges named skip connections which will connect previous states to current states. At the front we add some “D neurons” for which the current state is changed depending on previous state. Any state that occurred D time step ago will become a function of T/D, this will vanish or explode the state at D and not just a function of . The popular resonant architecture is precisely the implementation of this idea which takes place in CNN workplace.

### 3.9.2 Remove connection length

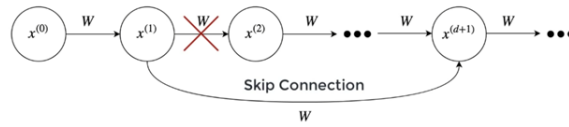


Figure 3.8: Remove connection length

This connection is created by actively replacing the connections that are of length 1 with connections that are lengthier. The modified path chooses what the network will learn.

### 3.9.3 Leaky recurrent Units

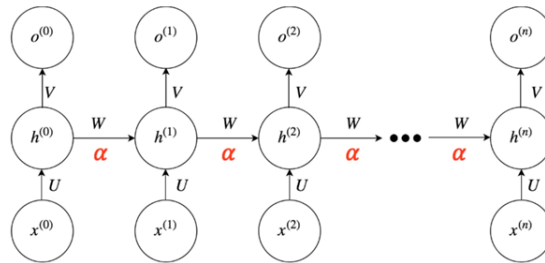


Figure 3.9: Leaky recurrent Units

It is considering the vanilla recurrent network instead of the modified recurrent neural network, hereby the consecutive hidden units are joined and a constant value of alpha is assigned at every edge. Over time the alpha value will determine the amount of information the network will hold or forget. The value of alpha if closer to 1 the memory is kept but if it is nearer to zero, the previous state memory is forgotten or removed.

### 3.9.4 Gated Recurrent Networks

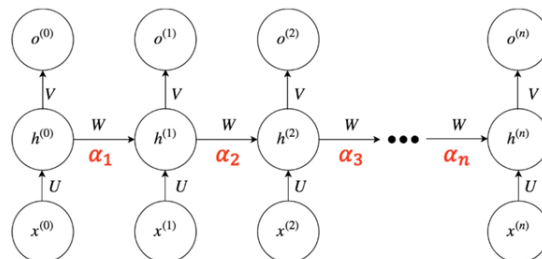


Figure 3.10: Gated Recurrent Networks

The gated recurrent networks are an extension of the leaky hidden units. In leaky recurrent units we need to assign a value alpha to decide what to keep. This value

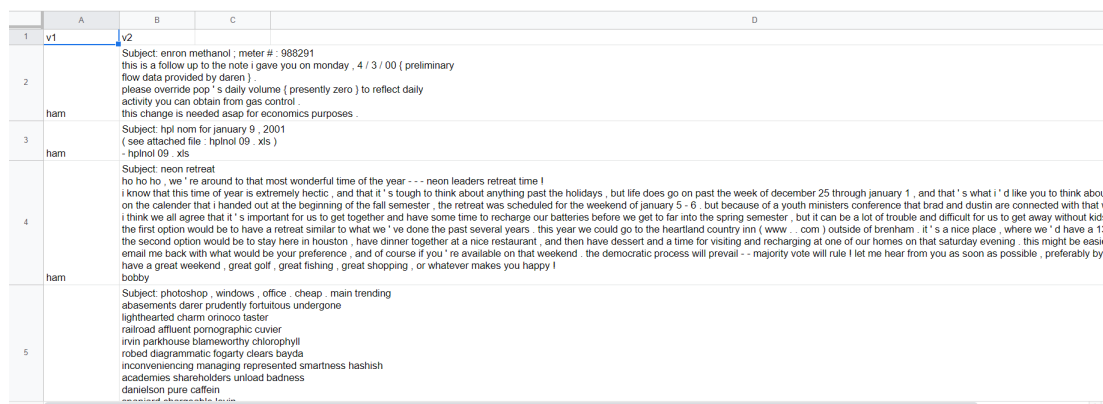
was manually assigned. At every time step new sets of parameters are introduced. Then the gated network can decide which parameters to keep or forget. The newly introduced parameters act as gates.

# Chapter 4

## Data and Model Workflow

### 4.1 Dataset

The dataset we are using is a part of the famous Enron Corpus which is a huge corpus containing more than 600,000 emails that has been made public during the legal investigation of Enron Corporation. But to save time and resources we have decided to train our selected model with a chunk of the Enron corpus, as the trained model is going to have almost the same efficiency. Our dataset contains 5,171 labeled emails with subject lines, Cc, Bcc to everything. In the csv file we mainly have 2 columns labeled as V1 and V2. V2 column consists of raw emails and V1 column contains the information that the corresponding email is spam or ham. As we have raw emails in our dataset it needs to go through a heavy cleaning process and the remaining ones will be sent to training and testing. The main Enron Corpus is located on the web at <http://www2.aueb.gr/users/ion/data/enron-spam/>. So here, our main goal is to train our model with supervised learning to detect spam emails. The datatypes of our columns are string/text and the V1 column can also be seen as a categorical value. This dataset has been chosen to fulfill some specific needs and there are not many open source email datasets out there that can be used for supervised learning. The "preprocessed" subdirectory provides pre-processed emails for training our model. There is a different text file for each post. The number at the beginning of each filename is the "arrival order".



	A	B	C	D
1	v1	v2		
2	ham	Subject: enron methanol ; meter # : 988291 this is a follow up to the note i gave you on monday , 4 / 3 / 00 ( preliminary flow data provided by daren ) . please override pop ' s daily volume ( presently zero ) to reflect daily activity you can obtain from gas control . this change is needed asap for economics purposes .		
3	ham	Subject: hpl nom for january 9 , 2001 ( see attached file : hplnol 09 . xls ) - hplnol 09 . xls		
4	ham	Subject: neon retreat ho ho ho , we ' re around to that most wonderful time of the year - - - neon leaders retreat time ! i know that this time of year is extremely hectic , and that it ' s tough to think about anything past the holidays , but life does go on past the week of december 25 through january 1 , and that ' s what i ' d like you to think about on the calender that i handed out at the beginning of the fall semester , the retreat was scheduled for the weekend of january 5 - 6 . but because of a youth ministers conference that brad and dustin are connected with that we i think we all agree that it ' s important for us to get together and have some time to recharge our batteries before we get to far into the spring semester , but it can be a lot of trouble and difficult for us to get away without kids the first option would be to have a retreat similar to what we ' ve done the past several years . this year we could go to the heartland country inn ( www . . . com ) outside of brentham . it ' s a nice place , where we ' d have a 13 - the second option would be to stay here in houston , have dinner together at a nice restaurant , and then have dessert and a time for visiting and recharging at one of our homes on that saturday evening . this might be easier email me back with what would be your preference , and of course if you ' re available on that weekend , the democratic process will prevail - - majority vote will rule ! let me hear from you as soon as possible , preferably by t! have a great weekend , great golf , great fishing , great shopping , or whatever makes you happy ! bobby		
5		Subject: photoshop , windows , office , cheap , main trending atassments dater prudently fortuitous undergone lighthearted charm orinoco taster railroad affluent pomographic cuvier irvin parkhouse blameworthy chlorophyll robed diagrammatic fogarty clears bayda inconveniencing managing represented smartness hashish academies shareholders unload badness danielson pure caffeine		

Figure 4.1: Enron1 parsed to CSV

## 4.2 Data Preprocessing

Data preprocessing is basically cleaning the raw data in an efficient and useful format so that unnecessary attributes can not affect the outputs and also it is a data mining technique. We look at our dataset in data preprocessing and determine if any null/nan values should be replaced. Then we need to remove the categorical variables into numeric form and replace them. If there are objects like strings in the NumPy array, we need to encode so that the matrix multiplications during the model fitting are effective. We also look for a duplicate value that might cause trouble in calculations. This is useful for making the dataset more accurate. It will help to gain a good accuracy score. Better accuracy indicates better performance of the dataset. In our dataset, we followed two fundamental steps to preprocess it which are cleaning the raw data and tokenized that clean data. For cleaning the raw data, we followed some standing rules so that the unnecessary words can be deleted. The steps are removal of hyperlinks, lowering case, removal of numbers, removal of punctuation, removal of whitespaces, replacing newline and cleanup pipelines. We remove all the URLs from the dataset and there is a chance that some of the emails can contain URLs. We do not want unnecessary value in our results. We also lower the case of all the data as it will reduce the dimension by decreasing the size of the vocabulary. And also, some data may have the same value in a sentence like ‘I’, ‘i’, ‘YOU’, ‘you’. So, that is why we lower the case of all data. We remove numbers from our datasets and remove all the punctuations from the dataset for example ‘who!’ in the given word we remove the exclamation mark (!). Then, we remove all whitespaces from the dataset and also replace newlines. At last, we use cleanup pipelines to run all above the functions we mention here.

## 4.3 Tokenization of the Cleaned Data

In machine learning algorithms, we need tokens as features. That is why, we process the whole email and split it into small chunks of words which is also known as tokenization. Keras has this built-in function to pre-process and tokenize data which keep the words that have the most number of occurrences in the data set. After tokenizing the email, we get a large bag of words which won’t all be necessary. To solve this problem, we can use ‘max\_features’ which will select the most frequently used unique words.

## 4.4 Sequencing

Text sequencing consists of two major steps which includes padding and the labeling of the encoding target variable. We can’t work with the different sized inputs as information might be lost in the process. So, we need padding to make the inputs of the same size. We use ‘max\_len’ to consider the length of all post-padding tokenized mails. After that, we need to convert our target variable to a number as the model will expect the target variable as a number and not a string. For that, we import built-in LabelEncoder from sklearn.

## 4.5 Model Selection

### 4.5.1 LSTM

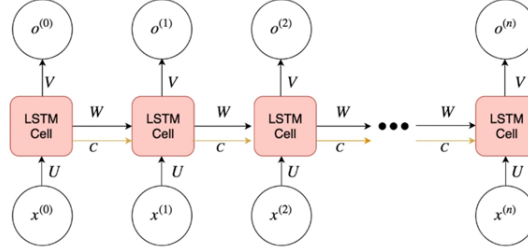


Figure 4.2: LSTM

Among all the gated recurrent neural network architectures is LSTM (long short-term memory) is the most popular and common. Considering a vanilla recurrent neural network and replacing all LSTM cells instead of the hidden units of the RNN structure. Consecutively updating every cell state with a connection from each individual cell. This way LSTM is formed. LSTM was intended to solve the problems of exploding gradient and vanishing gradient problems. Every LSTM cell retains a cell state vector including the hidden state vectors. The consecutive LSTM cell may choose to reset, read or write from the cell at every time step. A LSTM cell can be described by using a specific gating mechanism. Each LSTM unit is made up of gates. These gates are binary in nature usually they have three gates.

#### Input Gate:

$$i^{(t)} = \sigma(W^i [h^{(t-1)}, x^{(t)}] + b^i)$$

The memory cell state is updated through this gate.

#### Forget Gate:

$$f^{(t)} = \sigma(W^f [h^{(t-1)}, x^{(t)}] + b^f)$$

As the name suggests the forget can reset the value of the memory cell and it can reset it to 0.

#### Output Gate:

$$o^{(t)} = \sigma(W^o [h^{(t-1)}, x^{(t)}] + b^o)$$

It is the function of the output gate to decide the visibility of the current cell state to the next. All gates have regulated the model's differentiability by applying sigmoid function to it. Sigmoid function ranges from 1 to 0 and smoothens the curve of the functions.

$$\bar{C}^{(t)} = \text{tanH}(W^C [h^{(t-1)}, x^{(t)}] + b^C)$$

These are some of the gates. Excluding these gates consisting of the tanH function is able to modify the cell state. The function tanH has a range from -1 to 1, this operation helps to spread out the values of gradient correctly. This vanishing or exploding gradients problem is mitigated by this as the cell state is retained throughout the flow of data towards the next cell.

### How does an LSTM Function?

The current state input  $x^{(t)}$  and the previous hidden state input  $h^{(t-1)}$  goes into the LSTM cell. These two inputs are put into a sigmoid function after being concatenated, here  $\bar{C}$  is a candidate value which goes into the cell state. Gates are applied, as mentioned:

$$C^{(t)} = f^t C^{(t-1)} + i^{(t)} \bar{C}^{(t)}$$

The output of the input gate goes through a sigmoid function which gets a value to represent whether the input should be updated. This is then updated with  $\bar{C}$  so that the cell state can be updated to the new vector values. The Old state after passing through the forget gate can be forgotten or remembered depending on the output of the forget gate, this is then applied to the values of the output gate which in terms generates the hidden vector.

$$h^{(t)} = \tan H(C^{(t)}) \times o^{(t)}$$

### 4.5.2 GRU (Gate Recurrent Unit)

GRU is an updated version of LSTM that doesn't have memory or cell state. Only two gates are present namely update gate and reset gate. Although GRU has lesser gates it's proven through experiments that the results in exhaustive training GRU performs better than LSTM.

**Update Gate:**

$$z^{(t)} = \sigma(W^z[h^{(t)}, x^{(t)}] + b^z)$$

**Reset Gate:**

$$r^{(t)} = \sigma(W^r[h^{(t)}, x^{(t)}] + b^r)$$

### 4.5.3 Bi-LSTM

This model consists of two lstm that individually parses two sequences of data in both directions one through forward and the other through backward direction, thus it is called bidirectional LSTM or Bi-LSTM in short. Bi-LSTM can remember longer sequences of data and also predict text. It increases the capacity of holding data as it is parsed in both directions, it also has a context of what the actual data is about.

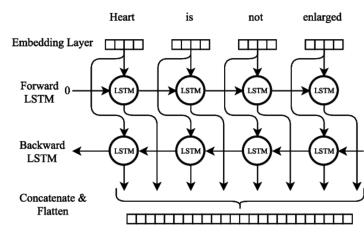


Figure 4.3: Bi-LSTM



## 4.6 Implementation

In our work we proposed a neural network that has an initial embedding layer of 32 nodes, which also acts as our input layer. The input layer takes in the pre processed data which is tokenized and padded. This data is passed through the input layer and moves to the first hidden layer which is actually an LSTM/Bi-LSTM or GRU layer containing 64 nodes, next the data is transferred to another hidden layer. This layer consists of 16 nodes and also the activation function of relu(Rectified Linear Units). We have set the dropout rate to 0.1 for better randomization of the input data. Finally the data enters the output layer consisting of 1 node and a sigmoid activation function. This gives us the output of the emails being spam or ham.

To implement the code, we need to first convert the formatted data into a machine understandable format. Basically, embedding is the process of converting pre-processed data into some numerical values or vectors so that a machine can easily interpret and analyze them. We are going to use some activation functions as we know without the activation functions the neural network will be less powerful and learning from complex dataset will be tough. That's why we need to use activation functions to get a better output for any kind of inputs. There are many kinds of activation functions like ReLU, Sigmoid, tanh, Leaky ReLU, Swish, Exponential Linear Unit and so on. We are using ReLU for our network as it helps us to speed up the training. It mainly works with 0 or 1 and also it activates one by one neurons rather than activating all of them together. The deactivation of the neuron will only happen when the value is less than 0. If there are any negative input values, the result will be 0 which means neurons will be deactivated. For this, the ReLU function is far more efficient to compute than Sigmoid or tanh. ReLU also allows us to converge quickly and allows backpropagation. We are also using another activation function to get the normalized output and the function is Sigmoid. To solve the overfitting problem easily, we are using dropout to randomize the data.

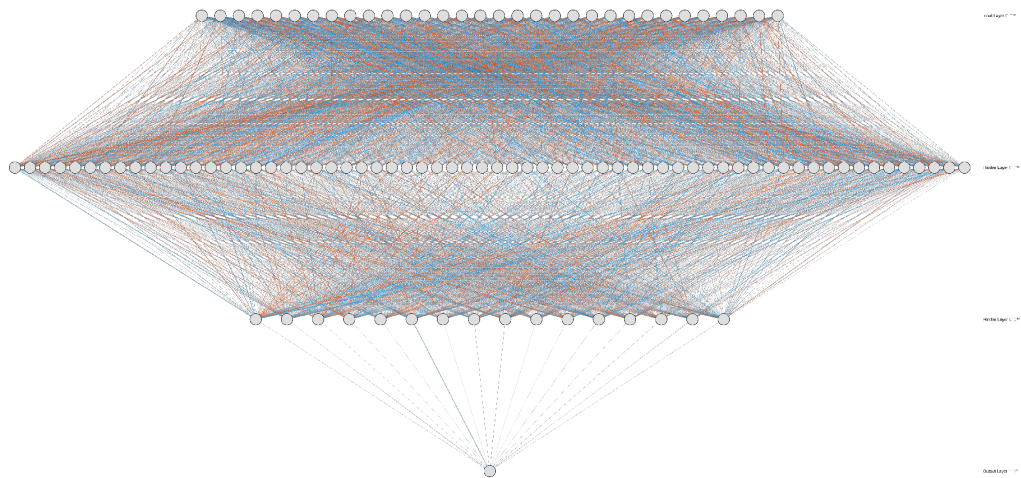


Figure 4.4: Proposed Neural Network Model

## 4.7 Workflow Diagram

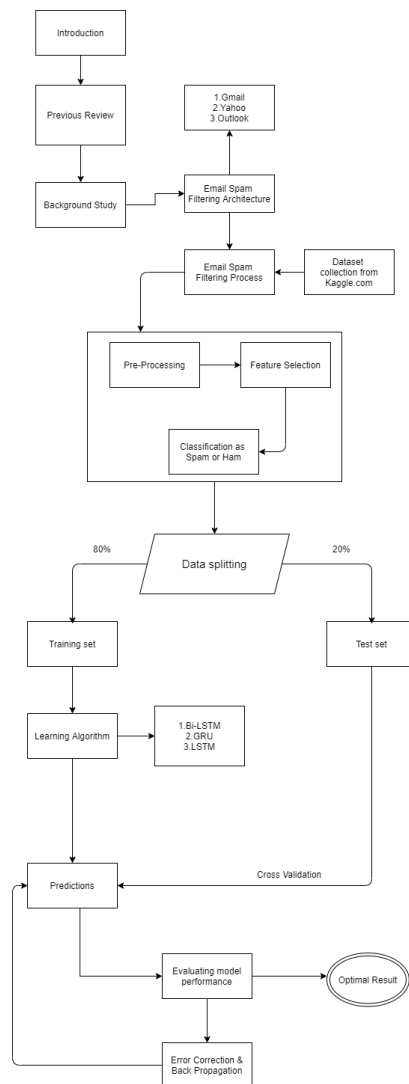


Figure 4.5: Workflow Diagram

# Chapter 5

## Result Discussion

### 5.1 Metrics of Evaluation

Precision, recall, accuracy, and F1 score are some of the standard metrics which are used for evaluation of the test results for or experiment.

These evaluation metrics are described below:

**Accuracy:** The no. of correct predictions is divided by total no. of predictions from the algorithms. Here,

$$Accuracy = \frac{(\text{True Positive} + \text{True Negative})}{(\text{True Positive} + \text{False Positive} + \text{False Negative} + \text{True Negative})}$$

**Precision:** The no. of positive predictions is divided by total positive valued class predicted from the algorithms.

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)}$$

**Recall:** The no. of positive predictions is divided by positive valued class predicted from the algorithms.

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)}$$

**F-Measure:** It's the measure of the stability of the ratio between recall and precision.

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

The confusion matrix constitutes the following metrics:

**True Positive (TP):** Prediction is positive and result is true.

**True Negative (TN):** Prediction is negative and result is true.

**False Positive (FP):** Prediction is positive and result is false.

**False Negative (FN):** Prediction is negative and result is false.

## 5.2 Model Accuracy and Confusion Matrix

### 5.2.1 Accuracy and Confusion Matrix LSTM

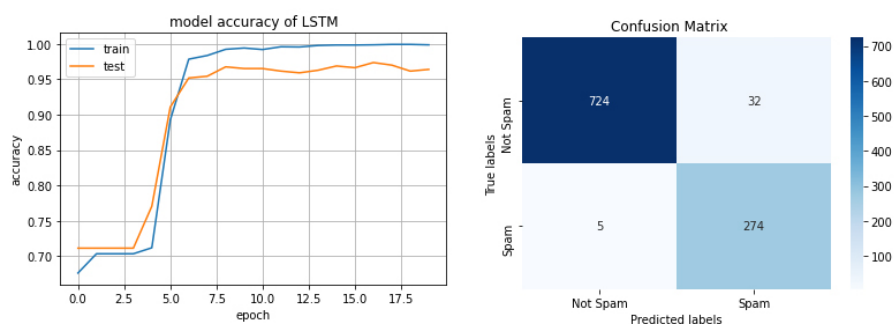


Figure 5.1: Model Accuracy and Confusion Matrix using Adam

From this figure, we can see the model accuracy graph of LSTM and also the confusion matrix for it. The model accuracy graph of LSTM shows the curve for training and testing of the dataset which seems quite similar. After that, the confusion matrix is showing that we get 274 spam and 724 non-spam mails from 1035 mails.

This figure is the graph for training and testing for our dataset. This model de-

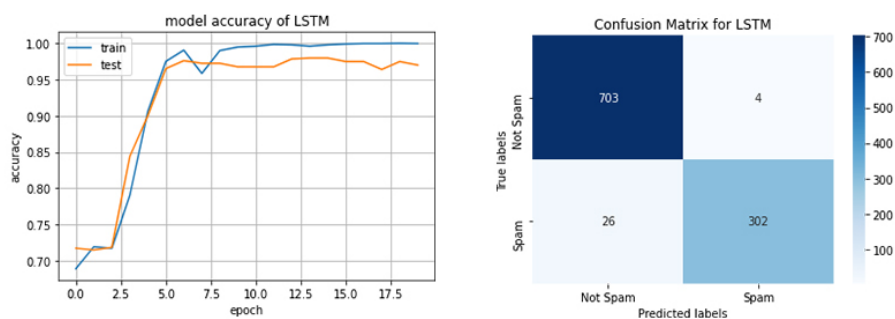


Figure 5.2: Model Accuracy and Confusion Matrix using Nadam

tects 703 non-spam and 302 spam among 1035 emails.

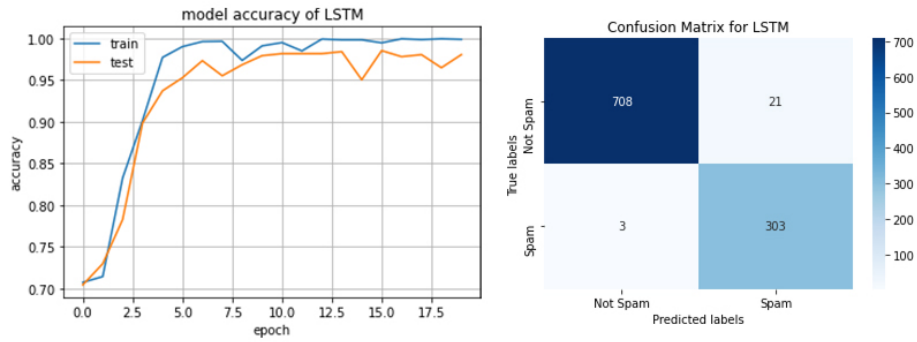


Figure 5.3: Model Accuracy and Confusion Matrix using RMSProp

From this figure, we can see the model accuracy graph of LSTM using rmsprop and also the confusion matrix for it. The model accuracy graph of LSTM shows the curve for training and testing of the dataset which seems quite similar. After that, the confusion matrix is showing that we get 303 spam and 708 non-spam mails from 1035 mails.

## 5.2.2 Accuracy and Confusion Matrix BiLSTM

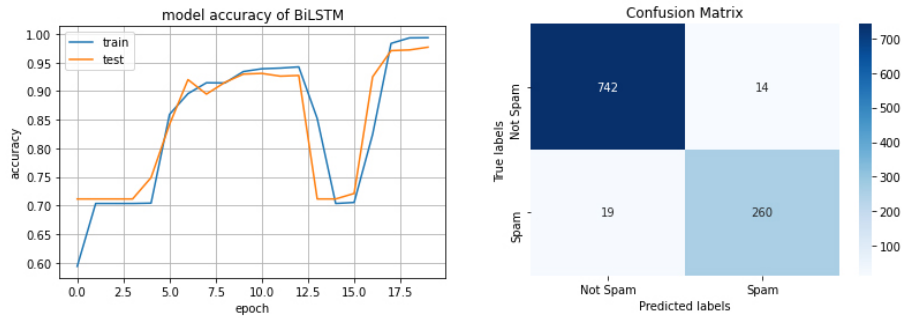


Figure 5.4: Model Accuracy and Confusion Matrix using Adam

From this figure, we can see that training data graph and testing data graph is quite similar which can be considered to be a good accuracy. On the other hand, this model is detecting 260 spam mails and 742 non-spam mails from 1035 emails.

From this figure, we can see that the graph of training data and the testing data

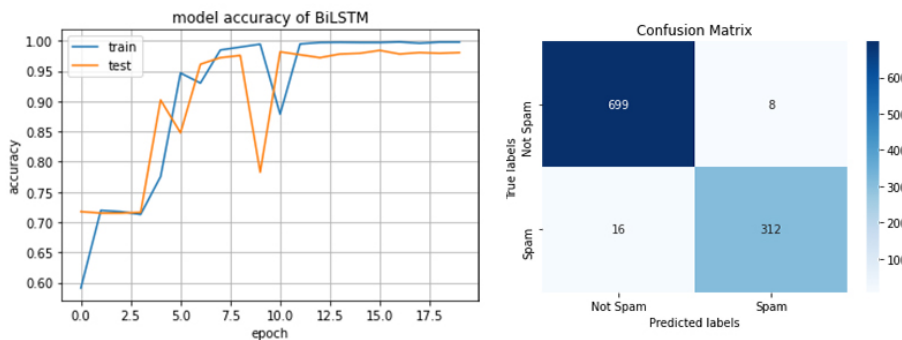


Figure 5.5: Model Accuracy and Confusion Matrix using Nadam

is quite similar. On the other hand, this model is detecting 312 spam mails and 699 non-spam mails from 1035 emails.

This figure represents the training and testing graphs for the dataset. Also this

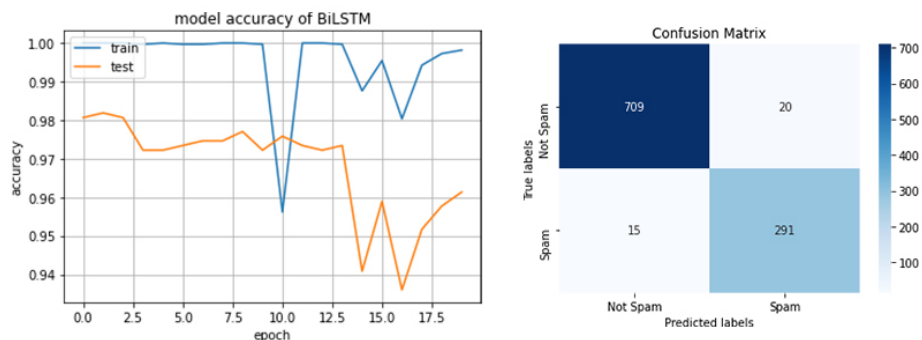


Figure 5.6: Model Accuracy and Confusion Matrix using RMSProp

model is detecting 291 spam and 709 non-spam from 1035 emails.

### 5.2.3 Accuracy and Confusion Matrix GRU

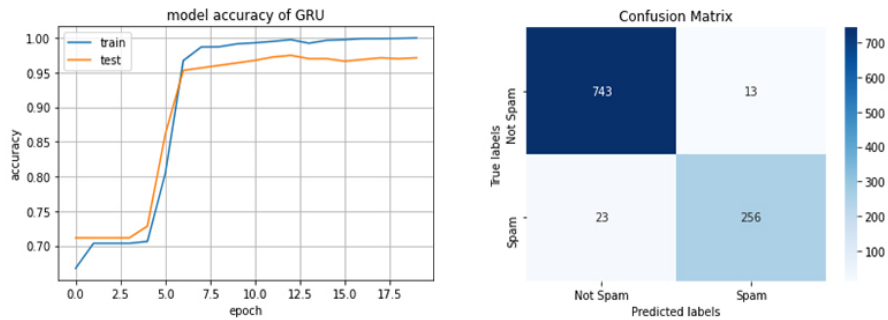


Figure 5.7: Model Accuracy and Confusion Matrix using Adam

This figure represents the training and testing graphs for the dataset. Also this model is detecting 256 spam and 743 non-spam from 1035 emails.

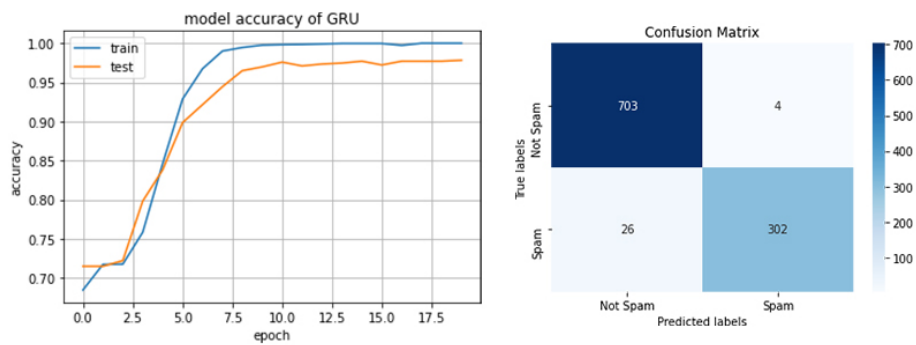


Figure 5.8: Model Accuracy and Confusion Matrix using Nadam

This figure represents the training and testing graphs for the dataset. Also this model is detecting 256 spam and 743 non-spam from 1035 emails.

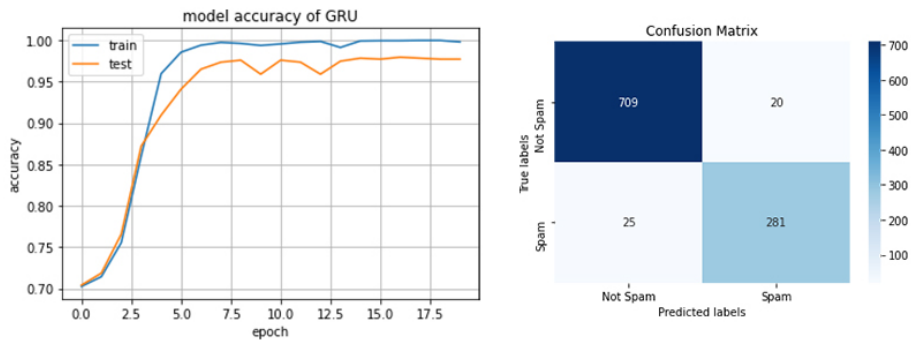


Figure 5.9: Model Accuracy and Confusion Matrix using RMSProp

This figure represents the training and testing graphs for the dataset. Also this model is detecting 281 spam and 709 non-spam from 1035 emails.

### 5.3 Output Analysis

If we look at the chart, we can easily see that it has used 3 models and 3 optimizers. We tried to use different methods to get an optimal result with our datasets.

**Adam Optimizer:** First of all if we look at the Adam optimizer for three methods which are LSTM, BiLSTM and GRU then we can see that there have been slight differences between value of Precision, Recall and F1-score. Precision value percentage of LSTM, BiLSTM and GRU for Adam optimizer is 89.54%, 94.89% and 95.17% and from the differences we can easily say that the GRU Precision is better than all of the methods. On the other hand, LSTM Recall value is greater than other meth-

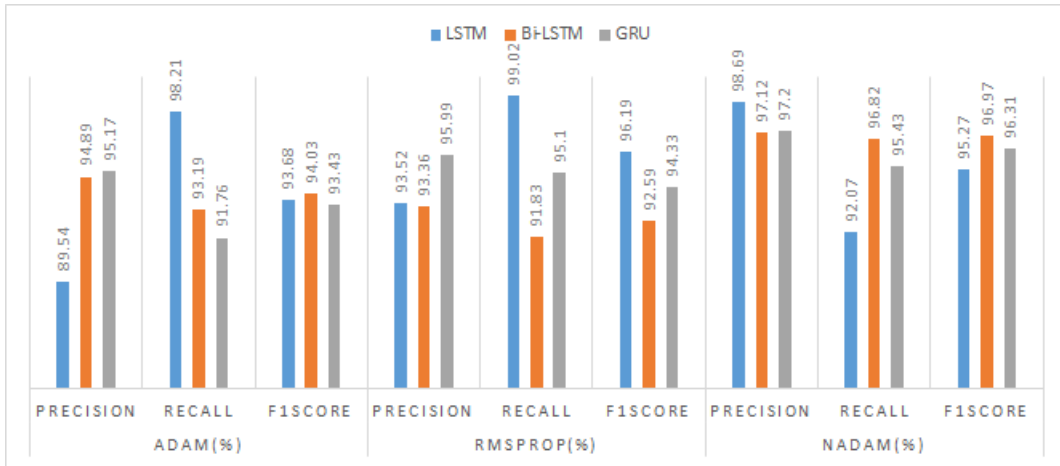


Figure 5.10: Output Analysis

ods which is 98.21% and BiLSTM and GRU's Recall values for Adam Optimizer is 93.19% and 91.76% and from these findings GRU's Recall value is lesser than other methods. Now, BiLSTM has the bigger F1-score value among all the methods for Adam optimizer which is 94.03%. But LSTM and GRU have the values of 93.68% and 93.43%.

**Rmsprop Optimizer:** We also get different values for Rmsprop optimizer for all the methods. GRU has the most value in Precision which is 95.99% and LSTM and BiLSTM have some slight differences between the Precision percentage and the values are 93.52% and 93.36%. Though in Adam optimizer LSTM has the least Recall value but in Rmsprop it has the highest value and it is 99.02%. BiLSTM has the least value of Recall in Rmsprop which is 91.83% and GRU has 95.99%. In F1-score of Rmsprop optimizer BiLSTM has 92.59% and it is the lowest value. BiLSTM got the bigger value for F1-score and the value is 96.19% and GRU has 94.33% value.

**Nadam Optimizer:** LSTM Precision value for Nadam optimizer is 98.69% and this is the highest value among LSTM, BiLSTM and GRU. Precision values of BiLSTM and GRU's are close to each other and they are 97.12% and 97.2%. Recall



values of Nadam optimizer for LSTM, BiLSTM and GRU are 92.07%, 96.82% and 95.43%. We can easily say BiLSTM has the largest value and LSTM has the least value. F1 score for Nadam optimizers is quite interesting as they have neck to neck values and BiLSTM has more value than LSTM and GRU which is 96.97%. LSTM F1-score value is 95.27% and this is the least value of F1-score for Nadam optimizer and GRU's F1-score is 96.31%.

From above all the findings and differences we can say that BiLSTM F1-score is best for Nadam optimizer among all the optimizers and we are giving focus mainly on F1-scores as using F1-using for detecting spam is a good choice. Precision and Recall is using for understanding the problem more accurately. Precision is the division between true positives and all the positives. On the other hand, Recall is going to help us know about how complete the positives are.

# Chapter 6

## Conclusion and Future Work

### 6.1 Future Work

The scope for interactive improvement is possible for our work in the future as AI is becoming more and more dominant in every field. Primarily the email archives from Enron, is insufficient for modern problems of spam. We can find a modified dataset that includes a complex dataset consisting of images and audio files. Secondly, the different datasets can include different neural networks such as CNN, RNN and LSTM. Thirdly, the inputs from these networks take different inputs and put them into a black box that generates the value as spam or ham. This model will not only segregate spam but also learn intricate patterns from the received spam for future predictions. Lastly, other variations of LSTM, RNN GRU together with or an individual hybrid RCNN-Bi-LSTM-GRU model could be put forward.

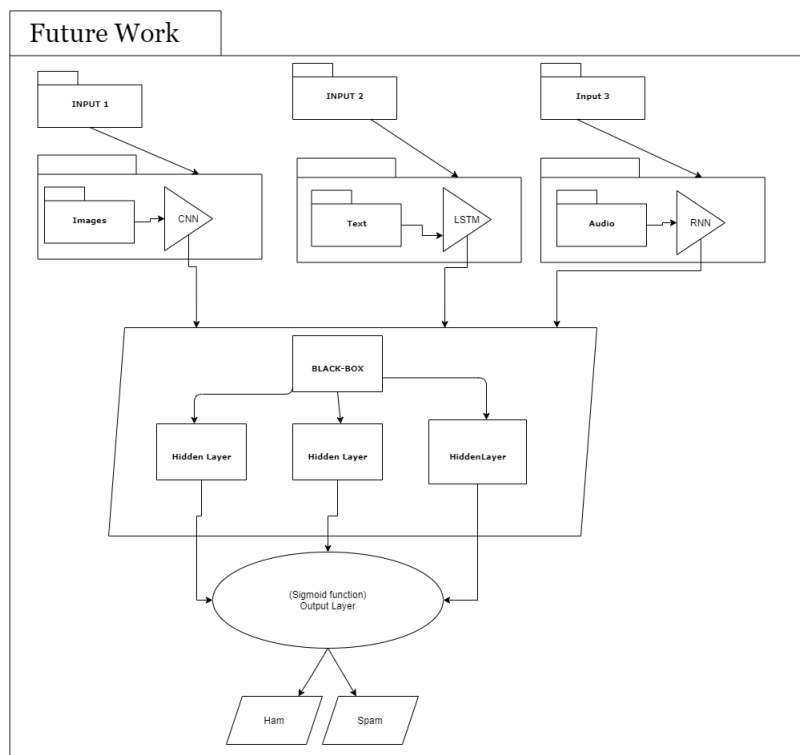


Figure 6.1: Proposed Future Model

## 6.2 Conclusion

Day by day, spam is becoming a serious problem for computer security as it becomes a main source for digital marketing, including viruses, disseminating threats, worms and phishing attacks. All the spammers are aware of the modern techniques and they try their best to bypass the spam detection system. The only way of dealing with these is to automate the system. Another horrific outcome would be that the spammer builds a neural network to bypass the existing neural network techniques. Currently, about 83% of received emails are spam. Many available measures to combat spams are email signature through SMTP, DKIM, SPF, DMARC and many third party software provides web mail archive software that can regulate mails coming into the server or going out. These solutions may not be as effective as ML or AI techniques, still many companies are using it. Many upscale software companies are struggling to deal with the spam situation but still it is prevalent due to the fact that the landscape of the web is shifting each and every day. So, no one can give a 100% surety that their approach is completely dependable for removing spam. That is why we proposed a system where we try to use Bi-LSTM, LSTM and GRU. We hope that our proposed model will perform better than the all previous work which has been done for detecting spam and also, we have outlined our work regarding this. Last but not the least, we think that a hybrid solution containing LSTM, CNN and RNN will help us to get maximum accuracy for detecting spam.

# Bibliography

- [1] B. Igelnik and Y.-H. Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.
- [2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [3] C. D. Rosin, R. S. Halliday, W. E. Hart, and R. K. Belew, “A comparison of global and local search methods in drug docking,” in *ICGA*, Citeseer, 1997, pp. 221–229.
- [4] G. P. Zhang, “Neural networks for classification: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.
- [5] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, “Stacking classifiers for anti-spam filtering of e-mail,” *arXiv preprint cs/0106040*, 2001.
- [6] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, “Multiclass alternating decision trees,” in *European Conference on Machine Learning*, Springer, 2002, pp. 161–172.
- [7] L. Pelletier, J. Almhana, and V. Choulakian, “Adaptive filtering of spam,” in *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, IEEE, 2004, pp. 218–224.
- [8] C. Dimitrakakis and S. Bengio, “Online adaptive policies for ensemble classifiers,” *Neurocomputing*, vol. 64, pp. 211–221, 2005.
- [9] M. R. Islam, M. U. Chowdhury, *et al.*, “Spam filtering using ml algorithms,” in *Proceedings of the IADIS international conference WWW/Internet 2005*, IADIS Press, 2005, pp. 419–426.
- [10] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. A. Baeza-Yates, “Link-based characterization and detection of web spam,” in *AIRWeb*, 2006, pp. 1–8.
- [11] G.-B. Huang, L. Chen, C. K. Siew, *et al.*, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [12] J. R. Mendez, F. Fdez-Riverola, F. Diaz, E. L. Iglesias, and J. M. Corchado, “A comparative performance study of feature selection methods for the anti-spam filtering domain,” in *Industrial Conference on Data Mining*, Springer, 2006, pp. 106–120.

- [13] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, “Detecting spam web pages through content analysis,” in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 83–92.
- [14] S. Youn and D. McLeod, “A comparative study for email classification,” in *Advances and innovations in systems, computing sciences and software engineering*, Springer, 2007, pp. 387–391.
- [15] J. Hu, Z. Li, Z. Hu, D. Yao, and J. Yu, “Spam detection with complex-valued neural network using behavior-based characteristics,” in *2008 Second International Conference on Genetic and Evolutionary Computing*, IEEE, 2008, pp. 166–169.
- [16] V. Christina, S. Karpagavalli, and G. Suganya, “Email spam filtering using supervised machine learning techniques,” *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 09, pp. 3126–3129, 2010.
- [17] L. Di Noi, M. Hagenbuchner, F. Scarselli, and A. C. Tsoi, “Web spam detection by probability mapping graphsoms and graph neural networks,” in *International Conference on Artificial Neural Networks*, Springer, 2010, pp. 372–381.
- [18] M. Mahmoudi, A. Yari, and S. Khadivi, “Web spam detection based on discriminative content and link features,” in *2010 5th International Symposium on Telecommunications*, IEEE, 2010, pp. 542–546.
- [19] A. Arram, H. Mousa, and A. Zainal, “Spam detection using hybrid artificial neural network and genetic algorithm,” in *2013 13th International Conference on Intelligent Systems Design and Applications*, IEEE, 2013, pp. 336–340.
- [20] K. L. Goh, A. K. Singh, and K. H. Lim, “Multilayer perceptrons neural network based web spam detection application,” in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, IEEE, 2013, pp. 636–640.
- [21] S. Singh, A. Chand, and S. P. Lal, “Improving spam detection using neural networks trained by memetic algorithm,” in *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, IEEE, 2013, pp. 55–60.
- [22] S. Shahane, S. Shendye, and A. S. Sinhgad, “Implementation of artificial neural network learning methods on embedded platform,” 2014.
- [23] Y. Zhang, S. Wang, P. Phillips, and G. Ji, “Binary pso with mutation operator for feature selection using decision tree applied to spam detection,” *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.
- [24] O. Fonseca, E. Fazzion, I. Cunha, P. H. B. Las-Casas, D. Guedes, W. Meira, C. Hoepers, K. Steding-Jessen, and M. H. Chaves, “Measuring, characterizing, and avoiding spam traffic costs,” *IEEE Internet Computing*, vol. 20, no. 4, pp. 16–24, 2016.
- [25] A. S. Katasev, L. Y. Emaletdinova, and D. V. Kataseva, “Neural network spam filtering technology,” in *2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, IEEE, 2018, pp. 1–5.

- [26] H. Faris, A.-Z. Ala'M, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, and H. Fujita, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks," *Information Fusion*, vol. 48, pp. 67–83, 2019.