SMS SPAM CLASSIFICATION

A thesis submitted to National Institute of Technology, Durgapur in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

NEHA SHARMA, MAHIMA KRITI, RUPAM MONDAL

Under the supervision of

Dr. BAISAKHI CHAKRABORTY

Associate Professor, Department of Computer Science and Engineering

National Institute of Technology, Durgapur

West Bengal – 713209

April, 2021

We dedicate this thesis to our loving family, friends, our project guide and research scholars and all the other people who have been a mentor and motivator during our project work and thesis completion.

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY,

DURGAPUR

WEST BENGAL – 713209

DECLARATION

We, Neha Sharma, Mahima Kriti, Rupam Mondal, hereby declare that the dissertation entitled "SMS SPAM CLASSIFICATION", submitted in the Department of Computer Science and Engineering, National Institute of Technology Durgapur, in partial fulfillment of requirements for the award of B. Tech. degree in Computer Science and Engineering, is an authentic work carried out by me during 2020-2021 under the supervision of Dr. Baisakhi Chakraborty.

Date: 13/04/2021
Place: Durgapur

(NEHA SHARMA, MAHIMA KRITI, RUPAM MONDAL)
Roll No:17IT8040, 17IT8016, 17IT8013
Department of Computer Science and Engineering
National Institute of Technology, Durgapur
West Bengal – 713209

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY,

DURGAPUR

WEST BENGAL – 713209

CERTIFICATE

It is to certify that the work contained in this thesis entitled "SMS SPAM CLASSIFICATION" has been carried out by Neha Sharma, Mahima Kriti, Rupam Mondal (Roll No. 17IT8040, 17IT8016, 17IT8013 respectively) and submitted for the award of the degree of Bachelor of Technology in Computer Science and Engineering. This work is bonafide research work under the guidance of Dr. Baisakhi Chakraborty. In my opinion, this thesis is of the standard required for the fulfillment of the requirement for the award of the degree of Bachelor of Technology.

Date: 13/04/2021
Place: Durgapur

Dr. Baisakhi Chakraborty
Associate Professor,
Department of Computer Science and Engineering
National Institute of Technology, Durgapur
West Bengal – 713209

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY,

DURGAPUR WEST BENGAL – 713209

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby approved as a credible study of a technological subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is to be understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn herein but approve the thesis only for the purpose for which it has been submitted.

_____ _____

Project Guide **Head of the Department**

_____ _____

_____ _____

Internal Examiners **External Examiners**

ACKNOWLEDGEMENT

We hereby wish to express my sincere gratitude and respect to Associate Prof. **Dr. Baisakhi Chakraborty**, Department of Computer Science and Engineering, National Institute of Technology Durgapur, under whom we had the privilege to work. Her valuable guidance and encouragement have really led us to the path completion of the project. Any amount of thanks would not be enough for the valuable guidance of our supervisor. It was thus that working under her expertise and receiving a part of her tremendous knowledge became a rewarding experience.

We would also like to thank all of the faculty members and research scholars of Department of Computer Science and Engineering for providing such a positive and healthy environment in the department.

At last, we would like to pen down our gratitude towards our parents, our family and our friends for their support, suggestions, and their constant encouragement. We would have been on a very rough journey without their constant support.

Date: 13-04-2021
Place: Durgapur

(NEHA SHARMA, MAHIMA KRITI, RUPAM MONDAL)
Roll No:17IT8040, 17IT8016,17IT8013
Department of Computer Science and Engineering
National Institute of Technology, Durgapur
West Bengal – 713209

ABSTRACT

SMS (Short Message Services) spam, which refers to an unsolicited message sent by a sender without prior relationship to the user mostly for commercial or financial purposes, is still a major problem to all Global System for Mobile communication (GSM) subscribers. The growth of the mobile phone users has led to a dramatic increase in SMS spam messages. Though in most parts of the world, mobile messaging channels are currently regarded as “clean” and trusted, on the contrary recent reports clearly indicate that the volume of mobile phone spam is dramatically increasing year by year. Most of the time such messages are commercial. But many times, such messages may contain some phishing links that have malware. This arises the need for proposing a prudent mechanism to detect or identify such spam messages so that time and memory space of the system can be saved up to a great extent. The present research emphasises to build a spam classification model with/without the use of ensemble of classifiers methods have been incorporated. Through this study, the aim is to distinguish between ham messages and spam messages by making an efficient and sensitive classification model that gives good accuracy with low false positive rate. In this paper, we presented the same mechanism which can filter spam and non-spam messages using two different machine learning methods and compared them both. Our proposed algorithm generates dictionaries and features and trains them through machine learning for effective results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As the utilization of mobile phone devices has become commonplace, Short Message Service (SMS) has grown into a multibillion dollars commercial industry. SMS is a text communication platform that allows mobile phone users to exchange short text messages (usually less than 160 seven-bit characters). It is the most widely used data application with an estimated 3.5 billion active users, or about 80% of all mobile phone subscribers at the end of 2010. As the popularity of the platform has increased, we have seen a surge in the number of unsolicited commercial advertisements sent to mobile phones using text messaging. However, the downside of the increase mobile users and the cheap SMS text messages is that mobile phones are attracting more unsolicited bulk messages especially in the form of advertisements. Compared to these unsolicited messages in SMS, spams commonly plague e-mails as 90 percent of the e-mails are spams in 2010. Although SMS spams are not as common as electronic junk mails, they still manage to irritate mobile phone users while creating societal frictions to mobile phone devices. Additionally, SMS Spam is particularly more irritating than email spams, since in some countries they contribute to a cost for the receiver as well. These factors along with limited availability of mobile phone spam-filtering software makes spam detection for text messages an interesting problem to look into. While, spams can be defined as "unwanted electronic mail". Spams are undesirable but still exist in our messages. SMS spams or mobile phone spams are junk mails delivered across mobile devices in the form of text messages. They are usually sent by spammers to intend a group of recipients by bulk. These spams usually sent by businesses taking advantages of receivers to advertise and promote their products or services. Besides promoting materials, spams also can threaten users' privacy with phishing, fraud and identify theft attacks through text messages. Spams can originate from any country in the world, with China topping other countries as the top source of spams. This shows that spammers do not refrain themselves from operating within their borders since some countries do little in preventing these spammers from spreading spams. Any individual can buy any mobile number from different area codes to spam mobile phone users; hence, they are hardly being identified and caught. Spam being a carrier of malware causes the proliferation of unsolicited advertisements, fraud schemes, phishing messages, explicit content, promotions of cause, etc.

Figure 1: Wordcloud For Ham

On an organizational front, spam effects include: i) annoyance to individual users, ii) less reliable messages, iii) loss of work productivity, iv) misuse of network bandwidth, v) wastage of file server storage space and computational power, vi) spread of viruses, worms, and Trojan horses, and vii) financial losses through phishing, Denial of Service (DoS), directory harvesting attacks, etc. Spam is a waste of time, storage space and communication bandwidth. Recently in the research community, the trend of classification using machine learning has become popular. As SMS message corpus have the tendency of growing bigger and complex along the time, a proper machine learning algorithm might be helpful to classify or filter the SMS Message spam characteristic. Machine learning approaches have been widely studied and there are lots of algorithms that can be used in SMS filtering. They include Naïve Bayes, support vector machines, Neural Networks, K-nearest neighbours, Rough sets and the artificial immune system. In this paper we will be using the Naive Bayes algorithm and RNN LSTM (Long short-term memory) to create two models that can classify dataset SMS messages as spam or not spam, based on the training we give to the model, and then compare accuracy of both the models. It is important to have some level of intuition as to what a spam text message might look like.

Figure 2: Wordcloud For Spam

Usually they have words like 'free', 'win', 'winner', 'cash', 'prize' and the like in them as these texts are designed to catch your eye and, in some sense, tempt you to open them. Also, spam messages tend to have words written in all capitals and also tend to use a lot of exclamation marks. To the recipient, it is usually pretty straightforward to identify a spam text and our objective here is to train a model to do that for us! Being able to identify spam messages is a binary classification problem as messages are classified as either 'Spam' or 'Not Spam' and nothing else. Also, this is a supervised learning problem, as we will be feeding a labelled dataset into the model, that it can learn from, to make future predictions.

CHAPTER 2

LITERATURE REVIEW

S. No	AUTHOR	PURPOSE	SOLUTION
1.	Gomatham Sai Sravya, G Pradeepini, Vaddeswaram, Guntur **- 2020**	To give the better accuracy	By using classification algorithms like Logistic Regression, Kneighbors Classifier, Random Forest Classifier, Decision Tree Classifier and Support Vector Machines
2.	Pradeep Kumar Roy, Jyoti Prakash Singh, Snehasish Banerjee **- 2020**	To filter SMS Spam efficiently	By using deep learning-based models such as CNN and LSTM along with machine learning based classifiers such as NB, RF, GB, LR, and SGD classifier
3.	Pavas Navaney, Gaurav Dubey, Ajay Rana **- 2018**	To detect better accuracy	By using various supervised machine learning algorithms like naïve Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm.
4.	Mehul Gupta, Aditya Bakliwal, Shubhangi Agarwal, Pulkit Mehndiratta **- 2018**	Evaluating machine learning techniques for spam SMS detection.	By comparing between traditional machine learning techniques (NB, SVM) and deep learning methods (CNN)
5.	Shafi'I Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul Salaam, Adamu I. Abubakar,	Detection and filtering of SMS Spams	By using different machine learning algorithms like SVM and Bayesian classifiers.

	Tutut Herawan **- 2017**		
6.	Lutfun Nahar Lota, B M Mainul Hossain **- 2017**	Increasing the Accuracy and decreasing the complexity	By having the SVM algorithm, it gives better accuracy but suffers from implementation complexity.
7.	Neelam Choudhary, Ankit Kumar Jain **- 2017**	For better accuracy	Used five machine learning algorithms namely Logistic Regression, Naive Bayes, J48, Decision Trees and Random Forest.
8.	Naresh Kumar Nagwani, Aakanksha Sharaff **- 2017**	Detecting the spam and nonspam messages	By using clustering techniques, we can detect the SMS spams and find the better accuracy
9.	Sakshi Agarwal, Sanmeet Kaur, Sunita Garhwal **- 2015**	To give the better accuracy	SVM and Multinomial Naive Bayes are used by having the dataset and calculated the accuracy for better score
10.	Dr. Ghulam Mujtaba, Majid Yasin **- 2014**	For better accuracy and performance	Used Naive Bayes classifier with hypertuned parameters to achieve better performance and accuracy
11.	Houshmand Shirani-Mehr **- 2013**	Reduce the spam messages and for better accuracy	By using UCI Repository dataset with different machine learning algorithms
12.	Kuldeep Yadav, Swetank K. Saha, Ponnurangam Kumaraguru, Rohit Kumra **- 2012**	To detect the SMS spam messages and calls	By using SVM, we can detect the SMS and give the better accuracy

CHAPTER 3

METHODOLOGY

3.1. ARCHITECTURE

The main objective of our approach is to classify the spam SMS messages received on the mobile phone. In this, the collected dataset is pre-processed and the features for our experiment are extracted using various mechanisms like Naive Bayes and RNN (LSTM unit). After the extracted features from the messages (ham and spam) create a feature vector. These feature vectors are used for training and testing purposes. Figure 1 and Figure 2 shows the system architecture of our proposed approach. In the training phase, a binary classifier is generated by applying the feature vectors of spam and ham messages. In the testing phase, the classifier determines whether a new message is a spam or not. At the end we get classification results for different machine learning algorithms and performance is evaluated for each machine learning algorithm such that we can get the best algorithm for our proposed approach.

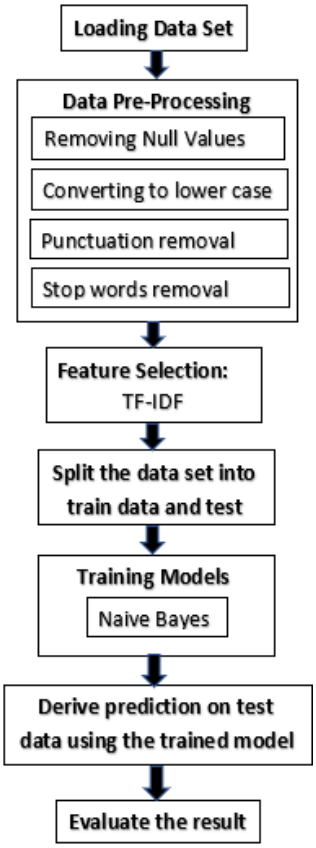

Figure 3: Architecture - Using Naïve Bayes

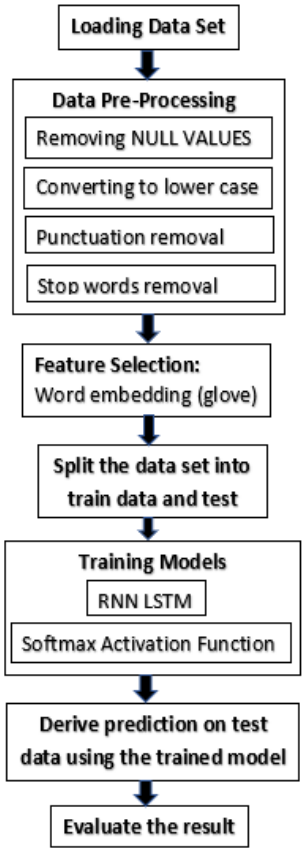

Figure 4: Architecture - Using LSTM Unit Of RNN

3.2. DATASET DESCRIPTION

The SMS (text) data that we have used for our experiment was downloaded from UCI datasets. It contains 5,574 SMS phone messages. The data were collected for the purpose of mobile phone spam research and have already been labeled as either spam or ham. The same dataset is used for both the applied methodologies. The dataset has 5574 rows and 2 columns. There are 4802 SMSs labeled HAM and 772 SMSs labelled SPAM in our dataset. Table 1 represent the dataset in more detail.

Serial No.	Label	No. of Entries
1.	Ham	4802
2.	Spam	772
		Total = 5574

Table 3: Distribution of Dataset

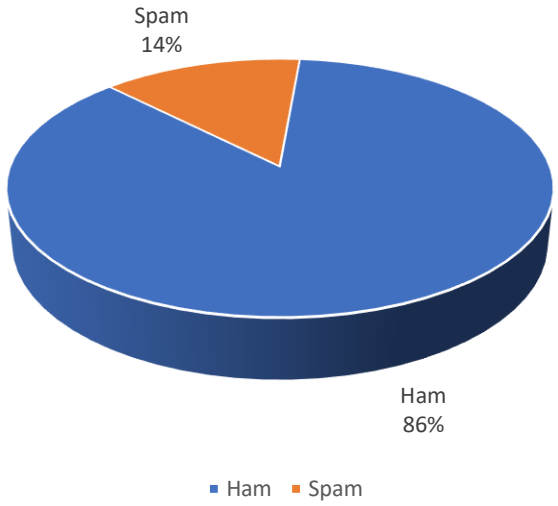

Figure 4: Pie-Chart Showing Distribution Of Dataset

3.3. DATA PRE-PROCESSING

Before starting with training, we must pre-process the messages. Different pre-processing techniques have been applied to different classifiers based on their requirement of input data.

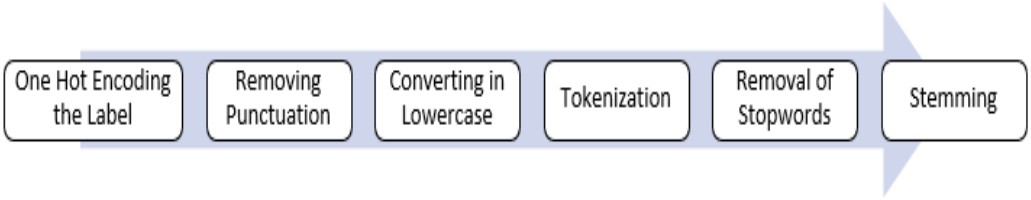

Figure 5: Different Ways Of Data Pre-Processing

Following is a brief description of these approaches.

1) One Hot Encoding: The SMS has been classified into 2 labels-ham and spam. Since scikit only deals with numeric values we convert our labels to binary variables,0 to represent ham (i.e., not spam) and 1 to represent spam for ease of computation. We use one hot encoding to achieve this.
2) Removal of all punctuation.
3) The SMS text is converted to lowercase.
4) Stemming: Words like ‘go’, ‘goes’, ‘going’ indicate the same activity. We can replace all these words by a single word ‘go’ which will reduce the number of new words encountered in the test dataset. This process is called stemming and we have Porter Stemmer a famous stemming algorithm for this purpose.
5) Removal of stop words: Stop words are those words which occur extremely frequently in any text. For example, words like ‘the’, ‘a’, ‘an’, ‘is’, ‘to’ etc. These words do not give us any information about the content of the text. Thus, it should not matter if we remove these words for the text.

Tokenization: Tokenization is the task of splitting up a message into pieces and throwing away the punctuation characters. Keras' Tokenizer is a class for vectorizing texts. It is used to turn texts into sequences. A sequence is a list of word indexes where the word of rank i in the dataset (starting at 1) has index i.

3.4. FEATURE EXTRACTION

Scikit-learn only deals with numerical values and hence if we were to leave our label values as strings, scikit-learn would do the conversion internally (more specifically, the string labels will be cast to unknown float values). Our model will still be able to make predictions if we leave our labels as strings but we could have issues later when calculating performance metrics, for example when calculating our precision and recall scores.

I. TF-IDF-Using Term Frequency - Inverse Document Frequency (tf-idf): In a given document, the count of the number of times a word appears is called Term Frequency. In a given corpus of documents, the number of times a word appears is called Inverse Document Frequency. Words are weighted according to the importance in tf-idf. Frequently used words have a lower weight, while words used infrequently have higher weight. Since we already started with removal of stop words, capital letters, non-alpha-numeric characters and any unnecessary punctuation. We then collected similar words (for example, desks will be transformed to desk). We then converted the cleaned text to tf idf features (5000 features for an entry) using sklearn's TfidfVectorizer to create a bag of words i.e., a count vector, followed by tf-idf matrix.

II. WORD EMBEDDING-with glove: Word embedding is one of the most popular representations of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. They are vector representations of a particular word. GloVe (Global Vectors for Word Representation) is an alternate method to create word embeddings. It is based on matrix factorization techniques on the word-context matrix. A large matrix of co-occurrence information is constructed and you count each "word" (the rows), and how frequently we see this word in some "context" (the columns) in a large corpus. Usually, we scan our corpus in the following manner: for each term, we look for context terms within some area defined by a window-size before the term and a window-size after the term. Also, we give less weight for more distant words. The number of "contexts" is, of course, large, since it is essentially combinatorial in size. So then we factorize this matrix to yield a lower-dimensional matrix, where each row now yields a vector representation for each word. In general, this is done by minimizing a "reconstruction loss". This loss tries to find the lower-dimensional representations which can explain most of the variance in the high-dimensional data. Glove is one of the most popular techniques to perform word embeddings using shallow neural networks and is preferred to

Word2Vec because GloVe does not rely just on local statistics (local context information of words), but incorporates global statistics (word co-occurrence) to obtain word vectors.

3.5. TRAINING & TEST DATASETS

To test our model, we split the data into train dataset and test dataset. We shall use the train dataset to train the model and then it will be tested on the test dataset. We shall use 75% of the dataset as train dataset and the rest as test dataset. Selection of this 75% of the data is uniformly random.

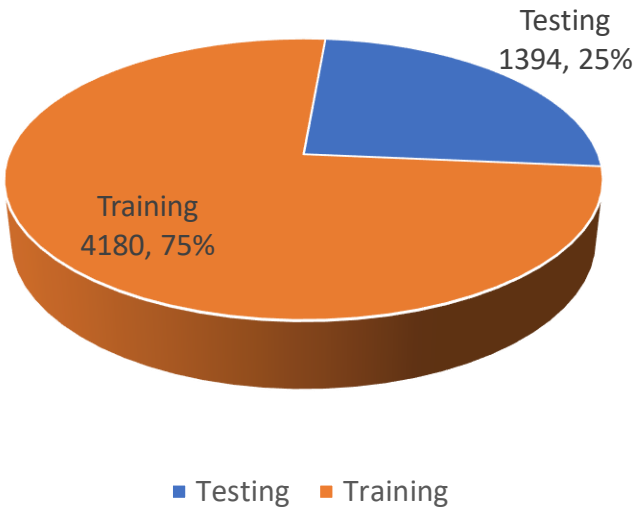

Figure 6: Splitting Of Datasets

3.6. DIFFERENT CLASSIFIERS

Two different pre-processing approaches have been applied to different classifiers based on their requirement of input data. Following is a brief description of these approaches.

I. Naive Bayes classifier - Naive Bayes is one of the simplest supervised learning algorithms. They have high accuracy and speed on large datasets. Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence. The Naive Bayes classifier

works on the principle of conditional probability, as given by the Bayes theorem. While calculating the math on probability, we usually denote probability as P.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- P(h): the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.
- P(D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.
- P(h|D): the probability of hypothesis h given the data D. This is known as posterior probability.
- P(D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability.

II. RNN with LSTM (Long Short Term Memory) units – A special architecture known as Long Short Term Memory (LSTM), a variant of the Recursive Neural Network (RNN) is used for spam classification. It has an ability to learn abstract features unlike traditional classifiers, where the features are hand-crafted. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell. The memory cell that has three logistic gates to control the output that goes to the next memory cell—the forget gate, the input gate, and output gate as shown in Fig. 4. These gates do not send their activities as input, instead they set the weights on the connections between the neural network and the memory cell. The memory cell has a self-connection which works as follows: When the forget gate has an activity of 1 i.e., it is turned on and the self-connection also has a weight 1, then the memory cell writes its contents to itself. When the forget, gate is set to zero, the memory cell discards its contents. The input gate is set to 1, it allows the rest of the

neural net to write into the memory cell and when the output gate is set to 1, the network can read from the memory cell. With the help of these three cells the LSTM is able to protect the network from exploding and vanishing gradient problem.

3.7. TESTING & EVALUATION

3.7.1. Performance Metrics

In order to evaluate the effectiveness of our proposed approach, we will consider eight possible outcomes i.e., true positive rate, false positive rate, true negative rate, false negative rate, f1 score, accuracy, precision, and recall. These are the standard metrics to judge any spam detection system. These evaluation metrics are described in brief as

follows:

- True Positive Rate (TP) - It denotes the percentage of spam messages that were accurately classified by the machine learning algorithm. If we denote spam messages as S and spam messages that were accurately categorized as P, then

$$\mathrm{TP} = \frac{P}{S}$$

- True Negative Rate (TN) - It denotes the percentage of ham messages that were accurately categorized as ham messages by the machine learning algorithm. If we denote ham message as H and ham messages that were accurately categorized as ham by Q, then

$$\mathrm{TN} = \frac{Q}{H}$$

- False Positive Rate (FP) - It denotes the percentage of ham messages that were wrongly categorized as spam by the machine learning algorithm. If we denote ham messages as H and ham messages that were wrongly classified as spam by R, then

$$\mathrm{FP} = \frac{R}{H}$$

- False Negative Rate (FN) - It denotes the percentage of spam messages that were incorrectly classified as ham message by the machine learning

algorithm. If we denote spam messages as S and number of SMS spam messages that were incorrectly classified as ham by T, then

$$\text{FN} = \frac{T}{S}$$

- Precision - It denotes the percentage of messages that were spam and actually classified as spam by the classification algorithm. It shows the exact correctness. It is given as -

$$Precision = \frac{TP}{TP + FP}$$

- Recall - It denotes the percentage of messages that were spam and classified as spam. It shows the completeness. It is given as –

$$Recall = \frac{TP}{TP + FN}$$

3.7.2. EXPERIMENTAL RESULTS

SMS unsolicited mail (every now and then known as cell smartphone junk mail) is any junk message brought to a cellular phone as textual content messaging via the Short Message Service (SMS). The dataset for this mission originates from the UCI Machine Learning Repository. Using special techniques to establish relation between the textual content and the category SPAM or HAM like, primarily based on length of message, word depend, unique keywords we have then constructed class fashions the use of one-of-a-kind strategies to differentiate spam SMS. To classify the text messages into Spam and Not-Spam, we tried in this paper two machine learning algorithms: (i) Naïve Bayes, (ii) a RNN (LSTM) algorithm. To train these classifiers, we used the same distribution of data for all algorithms: 75% for training and 25% for the test. We calculated three measures to compare the performance of the classifiers: Accuracy, Precision, Recall. Comparing the accuracy of each method and plotting the accuracy graphs in a single bar plot we see that RNNs perform better at spam classification than Naïve Bayes because of the inherent ability of RNN to process sequences.

Model	Precision	Recall	Accuracy
Naïve Bayes	88.73%	65.96%	94.16%
RNN (LSTM)	99.16%	98.75%	98.21%

Table 4: Performance Metrics Of The Dataset

From the table above, we can clearly see that the LSTM model is doing way better than the Naive Bayes Algorithm. The reasons can be due to:

- Tf-idf Vectorizer did not take into account the ordering of the words in the sentence, and it is losing a great deal of information.
- LSTM has been one of the greatest algorithms in sequence data (text, speech, time-series data) in the recent years.

Word embedding is definitely a good feature extraction tool for text data and with LSTM model, as we have built a spam filtering system with very decent performance.

CHAPTER 4

CONCLUSION AND FUTURE WORK

The SMS spam message problem is plaguing almost every country and keeps increasing without a sign of slowing down as the number of mobile users increase in addition to cheap rates of SMS services. This paper focused on how to filter SMS Spam efficiently. The main aim of this paper is to compare two different machine learning algorithms i.e., Naïve Bayes and RNN-LSTM with better accuracy score. The dataset that we have used in our work consists of 5574 messages which were collected from UCI datasets. The text messages are differentiated with Ham or Spam. It predicts whether the message in the dataset is Ham or Spam and predicts the performance through accuracy criterion. Accuracy chart is illustrated in the below bar graph.

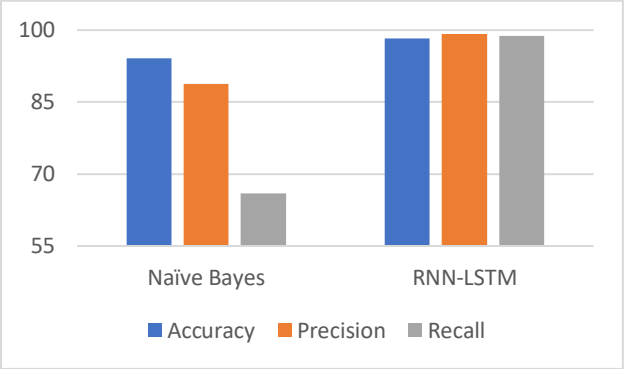

Figure 7: Comparison Of Classifiers

Therefore, we can safely conclude that building an SMS spam classifier using RNN-LSTM algorithm gives us the better results with an accuracy of 98.21% (Fig. 8).

A limitation of the work is that it was dependent on text messages written in English only. Therefore, this paper invites future research to employ similar deep learning approaches to filter Spam and Not-Spam text messages written in other languages too. Different algorithms will provide different performances and results based on the features used. So, for future work, adding more features such as message lengths might help the classifiers to train data better and give better performance We will also add more machine learning algorithm techniques to obtain best results.

CHAPTER 5

REFERENCES

i. Gomatham Sai Sravya, G Pradeepini, Vaddeswaram, Guntur, "Mobile Sms Spam Filter Techniques Using Machine Learning Techniques", International Journal of Scientific & Technology Research, 2020.

ii. Pradeep Kumar Roy, Jyoti Prakash Singh, Snehasish Banerjee, "Deep Learning to Filter SMS Spam", Future generation computer systems, Vol. 102, 01.2020, p. 524-533, 2020.

iii. Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya and Suriani Mohd Sam, "SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm", The Fifth Information Systems International Conference, 2019.

iv. Pavas Navaney, Gaurav Dubey, Ajay Rana, "SMS Spam Filtering using Supervised Machine Learning Algorithms", 8th International Conference on Cloud Computing, Data Science & Engineering, 2018.

v. Mehul Gupta, Aditya Bakliwal, Shubhangi Agarwal, Pulkit Mehndiratta, "A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers", Eleventh International Conference on Contemporary Computing, 2018.

vi. Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul Salaam, Adamu I. Abubakar, Tutut Herawan, "A Review on Mobile SMS Spam Filtering Techniques", , IEEE Access, 2017.

vii. Lutfun Nahar Lota, B M Mainul Hossain, "A Systematic Literature Review on SMS Spam Detection Techniques", International Journal of Information Technology and Computer Science, 2017.

viii. Neelam Choudhary, Ankit Kumar Jain, "Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique", International Journal of E-Services and Mobile Applications, 2017.

ix. Naresh Kumar Nagwani, Aakanksha Sharaff, "SMS spam filtering and thread identification using bi-level text classification and clustering techniques", Journal of Information Science, 2017.

x. Sakshi Agarwal, Sanmeet Kaur, Sunita Garhwal, "SMS spam detection for Indian messages", 1st International Conference on Next Generation Computing Technologies, 2015.

xi. Sakshi Agarwal, Sanmeet Kaur and Sunita Garhwal, "SMS Spam Detection For Indian Messages", 1st International Conference on Next Generation Computing Technologies 2015.

xii. O. O. Abayomi-Alli, S. A. Onashoga, A. S. Sodiya and D. A. Ojo, "A Critical Analysis Of Existing SMS Spam Filtering Approaches", Information Security Journal A Global Perspective, 2015.

xiii. Dr. Ghulam Mujtaba, Majid Yasin, "SMS Spam Detection Using Simple Message Content Features", International Journal of Scientific & Technology Research, 2014.

xiv. Houshmand Shirani-Mehr, "SMS Spam Detection Using Machine Learning Approach", 2013.

xv. Kuldeep Yadav, Swetank K. Saha, Ponnurangam Kumaraguru, Rohit Kumra, "Take control of your SMSes: Designing an usable spam SMS filtering system", IEEE 13th International Conference, 2012.