Journal of Positive School Psychology
2022, Vol. 6, No. 5, 7006–7013

SMS Spam Detection with Deep Learning Model

S. Nyamathulla[1], Polavarapu Umesh[2], Batchu Rudra Naga Satya Venkat[3]
challa Divya kumar[4]

[1,2,3,4] *Department of Information Technology, Vignan's Foundation for Science Technology and Research (Deemed to be University)*
Guntur, A.P, India
[1] *nyamath.j@gmail.com,* [2] *umeshpolavarapu71@gmail.com*
[3]*satyabatchu1127@gmail.com,* [4]*chinnachalladivyakumar@gmail.com*

Abstract

The number of mobile users is growing all the time, as well. SMS, which "short messaging service," enables users of both smartphones and conventional phones to send and receive text messages. Because of this, the number of SMS messages experienced a significant increase. In addition to that, the number of unwanted messages known as spam increased. The spammers' purpose is to distribute unsolicited electronic messages for commercial or financial gain, such as market penetration, the purchase of lottery tickets, or the disclosure of credit card information. As a direct consequence of this, sifting through spam receives additional attention. Several different machine learning and deep learning techniques, which are detailed in this Paper, were utilized to detect SMS spam. We developed a spam detection system based on data collected by the University of California, Irvine (UCI). This research study investigates the efficacy of several supervised machine learning algorithms, including the nave Bayes Algorithm, support vector machines, and the maximum entropy algorithm, in detecting spam and ham communications. Additionally, the outcomes of the detection of these messages are displayed here. SMS spam is becoming more prevalent as an increasing number of people use the Internet, and many enterprises share their personal information. SMS spam filtering inherits a substantial amount of functionality from e-mail spam filtering. When evaluating the effectiveness of various supervised learning strategies, the support vector machine method yields the most precise results.

Keywords— Short Message Service (SMS), Spam, Machine Learning (ML), Deep Learning (DL), LSTM, and UCI

INTRODUCTION

We primarily discussed and evaluated machine learning algorithms for detecting spam SMS. We compared eight different classifiers to each other. For both datasets, Convolutional Neural Network Classifier achieved the most remarkable accuracy of 99.19 percent and 98.25 percent, respectively, and the maximum AR value of 0.9926 and 0.99994. Yet, the traditional classifiers have not matched CNN's performance in text classification, even though it has been widely utilized in image classification. Because of CNN's success in text classification, a new avenue of investigation into issues like review categorization and sentiment prediction is now accessible to researchers[1].

The information and technological revolution have begun. As a result, many people are abandoning more conventional means of communication. Spam is a severe and vexing problem plaguing these information and communication methods. Our Modified Spam Detection Selection Data Set, which consists of Indian material, was subjected to various categorization procedures for analysis. The

Support Vector Machine and the Multinomial Naive Bayes were two of the most successful classifiers for identifying spam in SMS messages. Even though the SVM classifier equipped with a linear kernel achieved the highest level of accuracy, its computational requirements were too expensive. As an alternative, MNB with Laplace smoothing had a near-SVM-level accuracy but was much faster than SVM. We found that 98.23 percent of ACC percentage, 92.88 percent of SC percentage, and just 0.54 percent with SVM were the best findings from the Altered SMS Spam Collection Data Set with Indian content[2].

According to our findings, naive Bayes surpasses random forest and logistic regression when it comes to categorizing SMS spam. It was easy to determine if the text was spam or not using the Naive Bayes method, which has a high accuracy rate of 98.445 percent[3].

Naive Bayes and FP-Growth are superior to the average accuracy of each dataset when used in partnership. In this study, the algorithms employed for SMS categorization performed equally well, with an average accuracy above 90%. Using the spam Corpus v.0.1 Big SMS, it excels 1.154 percent, 0.025 percent on Spam Collection SMS, and 0.184 percent on the combined dataset. This accuracy is achieved by using the SMS Spam Collection v.1 dataset and applying the FP-Growth algorithm, which has an accuracy of up to 98.506 percent. 2. Due to SMS's limited number of characters, implementing minimal support alleviates issues related to limited features, which creates additional capabilities to distinguish between spam and ham SMS. 4. The FP-Growth may be used for datasets with a wide range of training data[4].

Improving the understanding of SMS text inputs is possible by using external knowledge sources (e.g., WordNet) rather than relying solely on static implicit information from the training data. Improved training efficiency without sacrificing performance is the goal of our approach. This study presents the Lightweight Gated Recurrent Unit, a new type of lightweight deep neural network model (LGRU). More than 30 current SMS spam classifiers were tested, and our model outperformed them all. An essential contribution of this research is that it exposes an approach that we hope can be applied to many different types of complicated recurrent models to reduce training complexity without sacrificing performance[5].

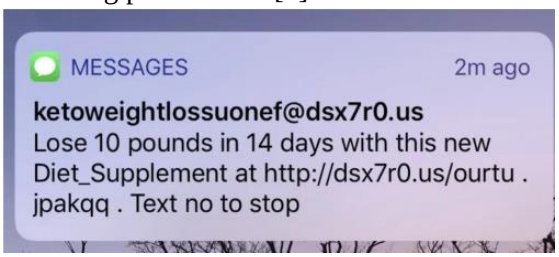


Figure 1: An example spam message proposing a weight loss regimen was shown above

The Telephone Consumer Protection Act (TCPA) was passed by Congress in 1991 to stop the unwanted meddling with consumers' lives through the transmission of unsolicited text messages, spam faxes, and other forms of unsolicited communication. The Telephone Consumer Protection Act (TCPA) addresses spam messages by instructing the Federal Communications Commission to regulate and prevent unwanted messages from being sent. The Federal Communications Commission employs various preventative measures to prevent SMS from transmitting spam messages, utilizing an auto dialler to mobile phones. There are a few exceptions, such as when the sender has the user's approval or when the communication is an emergency. In 2003, the CAN-SPAM Act, a new anti-spam regulation, came into being. This statute provides the same functions as the Telephone Consumer Protection Act, but it does so differently. The CAN-SPAM Act prohibits the transmission of advertisements or promotional text messages to mobile phones. When it comes to relationship communications, the CAN-SPAM statute does not ban them when they are linked to a product that the customer already owns. However, text messages from a new product or service that the consumer has never used fall into violation. For example, suppose a mobile user who has purchased a voltas air conditioner receives a promotional message from the Voltas corporation advertising upcoming deals. In that

case, this message is not considered spam by the anti-spam organization.

The CAN-SPAM Act and the Telephone Consumer Protection Act are similar. They both require vendors or website operators to obtain official approval before delivering any type of communication to customers. In contrast, according to the law, The CAN-SPAM Act can be violated if a random Godrej salesperson sends a promotional message regarding their air conditioning equipment.

BACKGROUND STUDY

It is feasible to handle the analysis and detection of spam messages by applying machine learning and deep learning strategies. We will train our naive Bayes classifier using data obtained from the SMS Spam collection, which is open to the public, has around 5574 items, and is free to use. The following is a list of researchers who have worked using the methods indicated above and the outcomes of their work.

Each SMS message in this dataset has its own label indicating whether the message was intended for or sent to someone else. Spam communications are labelled as such, whereas real blue messages are labelled as ham messages. The following illustration depicts a few examples of spam (Table 2) and ham (Table 1) in the context of a sandwich:

HAM MESSAGES

Table 1: Ham Messages

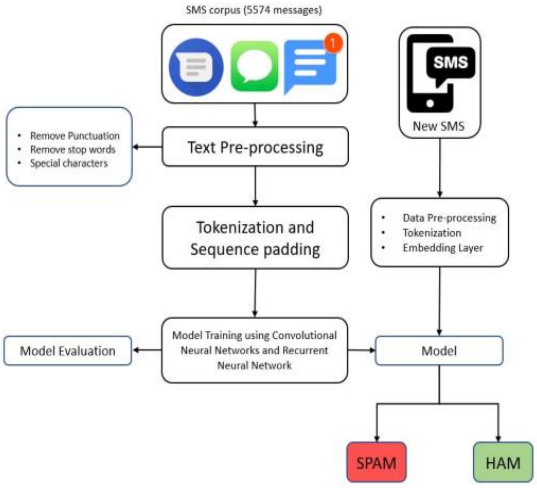
Make it realistic. And I'll see what happens.
Okay, I'll give it a shot, but I'm not sure I can commit.
My life is better as well. Yes, the weekdays are hectic due to work.

These are not a spam messages but rather communications that people regularly exchange. Therefore, the spam filter shouldn't prevent the client from receiving them.

SPAM MESSAGES

Table 2: Spam Messages

Shop for your favourite electronics at the Akshaya Tritiya Tech Fest and get up to 7.5% Instant Discount on HDFC Bank cards
Hi, strong credit scores qualify you for top loans and cards. 3 minutes to get your score
Open a digital account and let the game begins



The previous sample emails show some recognizable characteristics or repeating patterns of spam. The term "free" appears in two of the three spam communications but not in any of the ham messages. In contrast to the zero garbage communications, two of the spam messages mention certain days of the week. This is a fascinating discovery. Using word recurrence examples like these, our classifiers will determine if the SMS messages appear to be spam or ham based on their content. Even though "free" isn't unheard of outside of a spam SMS, a ham message will likely include other terms that describe the setting.

"Are you free on Saturday?" would be the question posed by a ham message instead of "Free tunes and ringtones," which can be the claim of spam. Each word in the message will serve as confirmation to the classifier, assessing if the message is spam or ham.

We have 5574 data, of which 4827 are genuine and 747 are spammers (Chart 1).

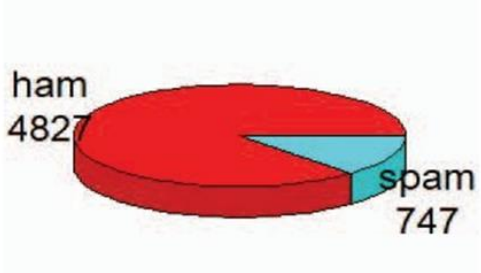


Figure 2: Pie chart of Spam Vs Ham

Data Set

Using the UCI repository, we obtained a dataset. There are 5572 rows and two columns in the dataset. There is a "spam" column and then a "ham" column. In this context, "spam" refers to an unsolicited message, whereas "ham" refers to a legitimate message. The news is found in the second column of the table.

Data Set	Training Data	Testing Data	Total
UCI	4746	986	5572

CLASSIFIER FRAMEWORK

The below figure 4 is the Proposed architecture for classifying spam and ham from 5574 SMS messages

	available	bugis	cine	crazy	got	great
1	Yes	Yes	Yes	Yes	Yes	Yes
2	No	No	No	No	No	No
3	No	No	No	No	No	No
4	No	No	No	No	No	No
5	No	No	No	No	No	No
6	No	No	No	No	No	No
7	No	No	No	No	No	No
8	No	No	No	No	No	No
9	No	No	No	No	No	No
10	No	No	No	No	No	No

Figure 3: Framework for Spam and Ham Classifier

We begin by importing the data into an excel record file from which we already have the raw data. An instant message is a sort of message which is either spam or ham. Type and message are both columns in our database. SMS messages are made up of words, punctuation, numbers, and breaks, all of which make up 160 characters. This level of detail necessitates a great deal of focus and effort on the user's part. We need to think about removing punctuation and numerals, stopping words like (and, or, but), and breaking up split sentences into single words, called fragments. Individuals from the R group have generously provided this tool in a "tm" text mining package.

Making a corpus and collecting text documents is a necessary first step in preparing content information. A text document, in this case, refers to a single SMS message. We are ready to create a sparse matrix data structure when we have removed all the stop words, punctuation marks, digits, and blank spaces from the text messages.

Table 3: Messages before and after cleaning

Preparing to Clean	After Having Cleaned
I hope your day is going well. merely confirming	Good luck, I was just checking.
Ok… Return my possessions.	In return, I'd want to thank you.
I see letter A on my bike	See letter A bike

After all of the information has been handled to our liking, the final step is to tokenize the messages. Tokens in this context are words, which are single components of a content string.

In the sparse matrix, each cell carries a number representing the number of words that appear in a specific phrase. The tokens are then represented in the form of a sparse matrix. In the sparse matrix, you can see which words are saved in the columns and stored in the rows. The DocumentTermMatrix, on its whole, has 5574 rows and 7958 columns, as shown in the accompanying screenshot(Fig.4).

	available	bugis	cine	crazy	got	great
1	Yes	Yes	Yes	Yes	Yes	Yes
2	No	No	No	No	No	No
3	No	No	No	No	No	No
4	No	No	No	No	No	No
5	No	No	No	No	No	No
6	No	No	No	No	No	No
7	No	No	No	No	No	No
8	No	No	No	No	No	No
9	No	No	No	No	No	No
10	No	No	No	No	No	No

Figure 4: Document Term Matrix

As can be seen from the table, several of the cells above are filled with "No," indicating that none of the terms listed above appear in the corpus's first 10 messages. As a result of this discovery, this data structure is a sparse matrix since most of the cells are filled with "No." No matter how many messages are sent, the odds of a specific term appearing in one are extremely low.The sparse matrix entry "yes" indicates that the words availability, Bugis, cine, crazy, got, and terrific appear in the initial text message.

A total of seventy-five percent of the messages were used for training, while just a quarter was

used for testing. The training dataset has 4171 records, while the testing dataset contains 1403.

Machine learning Techniques Incorporated

Subsequently, numerous machine learning classifiers, such as naive bayes, random forest decision trees, and so on, were developed. LSTM, a deep learning model, was also applied in our research activities.

Classification based on Naive Bayes

The Naive Bayes algorithm is a classification method constructed on the Bayes theorem's foundation. The concept of probability serves as the foundation for this theorem.

Logistic Regression

When dealing with classification tasks, Logistic Regression employs the logit and sigmoid functions to help you out. The output variable is predicted using an s-shaped curve as a guideline.

K-Nearest Neighbors

K-NN is a distance computation-based classifier that is both straightforward and efficient for use in machine learning. It finds the k neighbors that are the closest, then classifies the new data point according to the number of neighbors in the category that has the highest count.

Decision Tree Classification

The Decision Tree classifier builds a tree on which classification may be carried out so that the results can be analysed. After a specific number of minimum nodes has been reached, the tree is constructed repeatedly.

Random Forest classification

The Random Forest classifier considers the judgments of a significant number of decision trees before settling on categorizing a new data point. It is a method that emphasizes working together.

Support Vector Machine

A hyperplane is built during the SVM process based on which classification is performed. To locate the hyperplane, several datapoints are employed as support vectors**.**

LSTM

The term "neural network" encompasses both artificial and natural neural networks, including Recurrent Neural Networks (RNNs). The present state input of a Reinforcement Learning network may be determined using the network's prior state output. Recurrent neural networks commonly have problems with diminishing gradient descent, which makes them ineffective. In order to solve some of the shortcomings of standard RNNs, researchers have created RNNs with LSTM (Long Short-Term Memory). The researchers say that LSTMs are more suited to text mining problems than other methods.

Experimentation and results

Metrics for assessing

When comparing the performance of different classifiers and determining how successful they are, the three metrics utilized to do so are precision, recall, and accuracy.

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

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

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}$$

Outcomes of Count Vectorizer Research Results

The Logistic Regression model achieved the most incredible accuracy among the available options (95 percent). When it came to identifying the samples, we used six supervised machine learning techniques: Logistic Regression (with K-NN), Decision Tree (with DT), SVC (with SVC), Naive Bayes (with Random Forest), and Random Forest (with Random Forest). Table 4 presents the findings obtained from the examinations.

Table 4: Result of ML algorithm Experiments

Algorithm	Precision	Recall	Accuracy
Logistic Regression	91%	96%	95%
Navi Bayes	46%	89%	83%
Decision Tree	91%	97%	93%

SVM	94%	92%	91%
KNN	99%	45%	92%
Random Forest	94%	87%	90%

The data set is inconsistent in a variety of ways. Only 747 samples are allocated to the class label 1 out of 5572 pieces, which is a small number (spam). As a result, we adopted a sampling approach to ensure that the dataset was more evenly distributed. We relied on a methodology known as synthetic minority oversampling (SMOTE) to arrive at these findings. A heuristic approach to the generation of sample sets of data is what this method is. We can examine the tests carried out using the sampling method SMOTE by looking at Table 5. We were able to get a high level of accuracy by using logistic regression (95 percent).

Table 5: Experiments with a count vectorizer and SM

Algorithm	Precision	Recall	Accuracy
Logistic Regression	93%	99%	95%
Navi Bayes	95%	98%	93%
Decision Tree	87%	97%	90%
SVM	80%	99%	90%
KNN	65%	99%	92%
Random Forest	87%	98%	91%

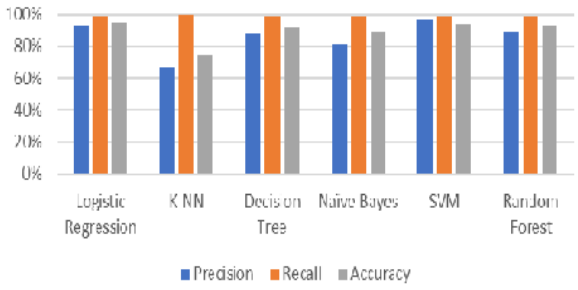


Figure 5: Results of Count Vectorizer and SMOTE

Experimentation with TF Shows Positive Results -IDF

The TF-IDF word embedding method was followed by six other categorization techniques. Experiments are summarized in table-6.

Table 6: TF Experiment Results -IDF

Algorithm	Precision	Recall	Accuracy
Logistic Regression	98%	73%	95%
Navi Bayes	52%	983%	86%
Decision Tree	87%	84%	96%
SVM	98%	83%	97%
KNN	84%	80%	84%
Random Forest	96%	82%	96%

The results of tests using TF-IDF are depicted in Figure 6. With SVM, we were able to attain the highest accuracy (97 percent). The accuracy of Decision Tree and Random Forest has also been shown to be greater than 96 percent.

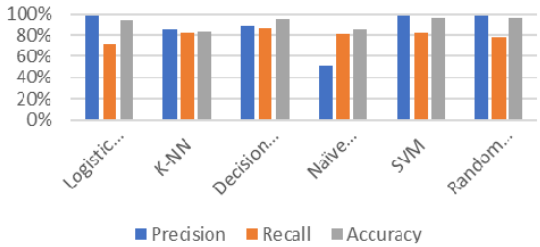


Figure 6: TF-IDF results of ML algorithms

Hashing Vectorizer Experiment: Results

Table 6: Experiments with TF-IDF

Algorithm	Precision	Recall	Accuracy
Logistic Regression	96%	72%	92%
Navi Bayes	30%	85%	65%
Decision Tree	84%	82%	92%
SVM	94%	82%	95%
KNN	94%	82%	86%
Random Forest	92%	84%	94%

Hashing Vectorizer word embedding and six classification methods were utilized. Table-6 shows the results obtained of the research. We achieved greatest accuracy using SVM (95 percent). Random Forest, on the other hand, has a 94% accuracy.

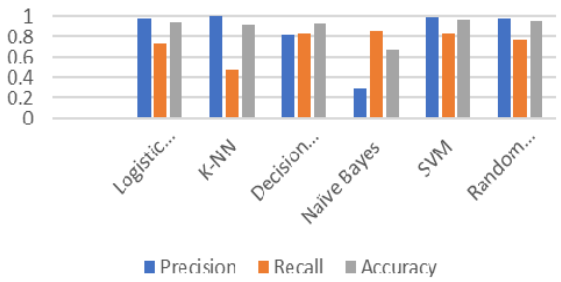


Figure 7: Hashing Vector results of ML algorithms

LSTM Results

After deploying all machine learning classifiers, we used a deep learning model to complete the task. We used LSTM to train a model on the same dataset and attained an accuracy rate of 98.5 percent.

Comparing Current Work to Previous Work

Table 7 compares the suggested model's accuracy with that of earlier research. In [6, the authors demonstrated that an ensemble model random forest classifier could attain an accuracy of 97.5 percent. With an SVM classifier, they reached an accuracy of 97 percent in [7]. According to our results, the accuracy of our suggested LSTM model was 98.5 percent.

Table 7: Examining Recent Work in Light of Prior Work

Type	Authenticity
Random Forest [6]	97.5
SVM [7]	97%
LSTM(Proposed)	98.5%

Conclusion & Future work

This paper aimed to create a machine learning-based deep learning model for identifying SMS spam. The information collected at the University of California, Irvine, was utilized in our research. Count vectorizer, TF-IDF, and Hashing Vectorizer were the three different word embedding strategies applied in this research. Following that, we classified the data using a variety of classification algorithms. The LSTM model provided an accuracy rate of 98.5 percent, which was satisfactory. The findings of the studies indicate that our model works better than earlier approaches to the detection of spam. Within the scope of this work, we only utilize one dataset. In the future, a diverse collection of datasets may be employed in order to implement the model.

REFERENCES

1. A. Alzahrani and D. B. Rawat, “Comparative Study of Machine Learning Algorithms for SMS Spam Detection; Comparative Study of Machine Learning Algorithms for SMS Spam Detection,” 2019.
2. S. Aluru, Jaypee Institute of Information Technology University, University of Florida. College of Engineering, IEEE Computer Society, IEEE Computer Society. Technical Committee on Parallel Processing, and Institute of Electrical and Electronics Engineers, 2018 Eleventh International Conference on Contemporary Computing (IC3) : 2-4 August 2018, Jaypee Institute of Information Technology, Noida, India.
3. S. Agarwal, S. Kaur, and S. Garhwal, “SMS spam detection for Indian messages,” in Proceedings on 2015 1st International Conference on Next Generation Computing Technologies, NGCT 2015, Jan. 2016, pp. 634–638. doi: 10.1109/NGCT.2015.7375198.
4. Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, “SMS spam detection using noncontent features,” IEEE Intelligent Systems, vol. 27, no. 6, pp. 44–51, 2012, doi: 10.1109/MIS.2012.3.
5. F. Wei and T. Nguyen, “A Lightweight Deep Neural Model for SMS Spam Detection,” Oct. 2020. doi: 10.1109/ISNCC49221.2020.9297350.
6. Nilam Nur Amir Sjarif, N F Mohd Azmi, Suriayati Chuprat, “SMS Spam Message Detection using Term Frequenct-Inverse Document Frequency and Random Forest Algorithm,” in The Fifth Information Systems International Conference 2019, Procedia Computer Science 161 (2019) 509–515,ScienceDirect
7. Pavas Navaney, Gaurav Dubey, Ajay Rana, “SMS Spam Filtering using Supervised Machine Learning Algorithms.,” in 8th International Conference on Cloud Computing, Data

Science & Engineering, 978-1- 5386-1719-9/18/ 2018 IEEE.

8. M. Nivaashini, R.S.Soundariya, A.Kodieswari, P.Thangaraj, ": SMS Spam Detection using Deep Neural Network.," in International Journal of Pure and Applied Mathematics, Volume 119 No. 18 2018, 2425-2436.
9. Gomatham Sai Sravya, G Pradeepini, Vaddeswaram, ": Mobile Sms Spam Filter Techniques Using Machine Learning Techniques.," Internatıonal Journal Of Scıentıfic & Technology Research Volume 9, Issue 03, March 2020.
10. S. Sheikhi,M.T.Kheirabadi,A.Bazzazi, "An Effective Model for SMS Spam Detection Using Content-based Features and Neural Network", : International Journal of Engineering, IJE TRANSACTIONS B: Applications Vol. 33, No. 2, (February 2020) 221-228.
11. A.Lakshmanarao,K.ChandraSekhar, Swathi, "An Efficient Spam Classification System Using Ensemble Machine Learning Algorithm," in Journal of Applied Science and Computations, Volume 5, Issue 9, September/2018.
12. Luo GuangJun,, Shah Nazir, Habib Ullah Khan, Amin Ul Haq, "Spam Detection Approach for Secure Mobile Messgae Communication using Machine Learning Algorithms.," in Hindawi, Security and Communication Netwroks,Volume 2020,Article id:8873639.July-2020.
13. Anju Radhakrishnan et al, "Email Classification using Machine learning algorithms", International Journal of Engineering and technology(IJET).
14. Shafi'l Muhammad Abdulhamid , "A Review on Mobile SMS Spam Filtering Techniques", IEEE Access, 2017.