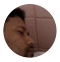
SPAM DETECTION MODEL USING TENSORFLOW AND DEEP LEARNING ALGORITHM

Article *in* Malaysian Journal of Computing and Applied Mathematics · September 2023

DOI: 10.37231/myjcam.2023.6.2.84

Mohd Fadzil Abdul Kadir
Universiti Sultan Zainal Abidin | UniSZA
56 PUBLICATIONS **231** CITATIONS



Mohamad A Mohamed
Universiti Sultan Zainal Abidin
163 PUBLICATIONS **1,434** CITATIONS

Nazirah Abd. Hamid
Universiti Sultan Zainal Abidin | UniSZA
23 PUBLICATIONS **68** CITATIONS

Siti Dhalila Mohd Satar
Universiti Sultan Zainal Abidin | UniSZA
19 PUBLICATIONS **69** CITATIONS

MALAYSIAN JOURNAL OF COMPUTING AND APPLIED MATHEMATICS (2023) 6 (2) 11-21
©UNIVERSITI SULTAN ZAINAL ABIDIN
DOI: *https://doi.org/10.37231/myjcam.2023.6.2.84*

Article: Received 18.11.2022; Accepted 16.06.2023; Published 30.09.2023

SPAM DETECTION MODEL USING TENSORFLOW AND DEEP LEARNING ALGORITHM

Shalini Govindan, Ahmad Faisal Amri Abidin, Mohamad Afendee Mohamed, Siti Dhalila Mohd Satar, Mohd Fadzil Abdul Kadir, Nazirah Abd Hamid

Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
shalinigovindan7@gmail.com

Abstract: As technology is becoming an integral aspect of every person's life since it makes living simpler and more efficient. As technology advances, spam attacks are becoming increasingly widespread. In this paper, we propose a solution to tackle this issue by comparing Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) deep learning techniques. Our objective is to develop a spam detection model that can effectively identify and filter out spam text. To evaluate the performance of our proposed model, we conducted experiments on a meticulously curated dataset consisting of spam and ham instances. The RNN model demonstrated exceptional accuracy, achieving an accuracy rate of 98.36%. Comparative analysis revealed that the CNN model achieved an accuracy rate of 97.10%, while the LSTM model attained an accuracy rate of 92.85%. These findings highlight the superior performance of the RNN model in accurately detecting spam using the TensorFlow platform. This research contributes to the advancement of spam detection methodologies, providing valuable insights for the development of effective spam filtering systems in the digital realm.

Keywords: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Spam Email

1. INTRODUCTION

In today's digital world, spam has become a widespread and bothersome issue. It refers to the unwanted and unsolicited bulk electronic communication that inundates our inboxes. It is getting harder to keep individuals from falling victim to spam because spam is on the rise everywhere in the world. This makes the efforts that are being made to do so more difficult. Email is the most common method for spam distribution. However, it can also be transmitted via text message, phone call, or shared via social media. Spamming can cause a range of harm, which has led to identity theft, the loss of private and personal information, and a lack of information assurance.

By using a spam detection system, the user's experience is improved due to its ability to identify unsolicited and unwanted emails and prevent spam messages from entering the user's inbox. Some detection methods are employed to address these problems; however, they are insufficient. The deep learning technique introduces an approach driven by datasets to achieve better classification and prediction between spam and ham.

To detect spamming, deep learning can be the most suitable method by using TensorFlow, which is a platform for developing machine learning applications that are open source and include all the available features. It assists in putting best practices into effect, such as those for data automation, model tracking, performance monitoring, and model retraining. Deep learning is One branch of machine learning that can be broken down further. It is a field that is constructed on the principle of self-improvement through the study of computer algorithms. Deep learning uses artificial neural networks, which are meant to mimic how humans learn and think. The use of deep learning algorithms in machine learning for spam detection represents a new and promising direction in this field, as it has the potential to improve the efficiency and accuracy of spam detection.

2. RELATED WORKS

The research carried out by [1] despite the abundance of security mechanisms in message sending, spam is one of the most common assaults customers experiences. Organizations incur substantial losses due to spam messages, which also raise mail server space, network bandwidth, spam filtering, and mail server processing. Spam requires users to waste more time and effort eliminating and cleaning the trash and discarding the contents of these unwanted messages.

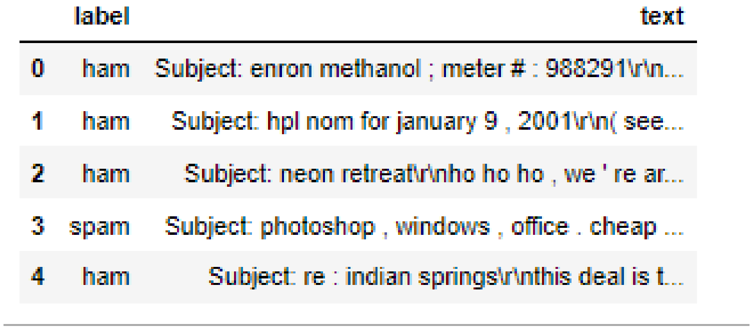
In the research paper, the email services are now one of the most prevalent forms of communication. However, spammers have multiplied in number. Spam encompasses all unwanted messages delivered to several users to transmit infections, steal users' privacy, and gain personal information. Numerous strategies, including deep learning algorithms, have been developed to distinguish spam from genuine e-mail. Using the most recent machine-learning technologies, this study aims to improve the accuracy of spam detection and prevention models. The six classifiers, including CNN with deep learning, is used to categorise the same Python-designed features with the same software. CNN is a 4D array that converts the array to 2D to match machine learning inputs with extracted features and best-selected features from real images, thereby feeding the classifiers utilised in suggested method, including CNN, KNN, RF, DT, SVM, and NB. Given the results, CNN remains the most accurate classifier. There are no notable modifications to the running time and performance. Also, the proposed method's classifier consistently produced the best results for all data split amounts. The proposed model achieved over 99% accuracy on spam image detection [1].

In the study conduct by [9], to buy stuff from internet sites without being cheated, a robust and dependable system for detecting spam reviews is necessary for the modern era. There are options for publishing reviews on numerous websites, creating opportunities for fraudulent paid or false reviews. The proposed model is divided into four phases. In the first step, both labelled and unlabelled datasets are acquired and pre-processed. All the unlabeled data are gradually labelled through the Active Learning Algorithm in the second phase. Techniques such as TFIDF, n-grams, and Word Embeddings (Word2Vec) are utilized in the third step of the feature selection process. The final phase is the spam detection phase, during which traditional machine learning classifiers and deep learning classifiers are employed to classify reviews as spam or ham. We used both labelled and unlabeled data and developed deep learning algorithms for spam review identification, including Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and a Long Short-Term Memory (LSTM) variation of Recurrent Neural Network (RNN). Furthermore, traditional machine learning classifiers such as Nave Bayes (NB), K Nearest Neighbor (KNN), and Support Vector Machine (SVM) were used to identify spam reviews [9].

In the research executed by [2] spam has long been an essential yet challenging subject to address. Researchers have created several machine learning-based algorithms and blocklisting techniques to detect spamming on Twitter. According to the investigation, current methodologies and techniques have approximately 80 percent accuracy. Due to the difficulties of spam drift and information manipulation, these machine-learning based approaches cannot detect spam behaviours in real-world scenarios. To solve the existing challenges, the research offered a novel technique based on deep learning techniques. The syntax of each tweet will be learned using Word Vector Training Mode, and then a binary classifier will be formed using the previous representation dataset. To evaluate the suggested strategy, the experiment collected and implemented 10-day actual Tweet datasets. Before comparing the method to other text-based methods, analyze the performance of various classifiers. Then, it was discovered that the method outperformed all other methods by a significant margin. After it, compare the strategy further to non-text-based detection strategies. Then, they utilized the Word Vector technique for pre-processing them and transforming them into vectors with a high dimension [2].

3. DATASET

The dataset used in this study was downloaded from Kaggle, a reputable website recognized for hosting high-quality datasets. The dataset includes 5,171 text samples that have been expertly labeled as either ham or spam. These labels are designated according to the nature and content of the messages. There are 3,672 instances labeled as ham, representing non-spam messages that are legitimate. Typically, these are standard text communications, such as personal emails or business letters. On the other hand, 1,499 instances are identified as spam, which consists of unsolicited and undesirable messages. These spam messages usually contain advertisements, phishing attempts, or fraudulent content. This dataset allows us to examine the characteristics and patterns of spam and non-spam messages (Figure 1).



	label	text
0	ham	Subject: enron methanol ; meter # : 988291\r\n...
1	ham	Subject: hpl nom for january 9 , 2001\r\n(see...
2	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...
3	spam	Subject: photoshop , windows , office . cheap ...
4	ham	Subject: re : indian springs\r\nthis deal is t...

Figure 1 Dataset ham and spam

4. PROPOSED METHOD

In this section, we propose a framework with multiple phases, including data acquire, data preprocessing, training, testing, evaluation, and retraining as shown in Figure 2. The system uses a dataset obtained from Kaggle, a trustworthy source of spam datasets. TensorFlow is used for data pre-processing, training, and testing, whereas deep learning algorithms such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) are employed for training the model. Applying evaluation metrics, the performance of the model is evaluated. In addition, a retraining and retesting phase is conducted using an unlabeled dataset to improve the spam detection system's accuracy and efficacy.

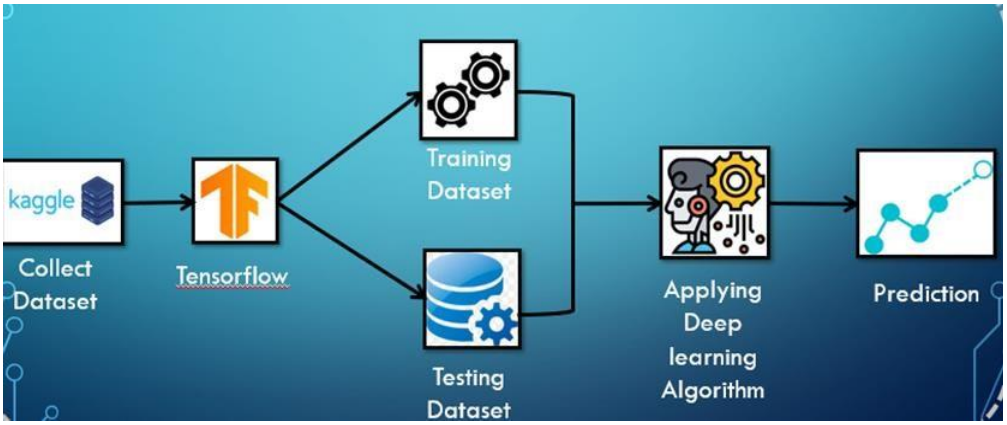


Figure 2 Framework of the model

Collect Dataset:
The first stage is to acquire involves the collection of spam datasets from Kaggle, a reliable source of datasets. These datasets are necessary for training and testing the effectiveness of the spam detection system.

Data Pre-processing:
The next stage focuses on data pre-processing using TensorFlow. This step involves includes preparing and arranging the collected data to allow for further analysis. TensorFlow provides various tools and capabilities to effectively process and manipulate data.

Training and Testing:
Once the data has been pre-processed, TensorFlow is used for training and testing. The collected dataset is separated into training and testing sets. The training set is used to teach deep learning algorithms, such as CNN, RNN, and LSTM, the patterns, and characteristics of spam messages. The testing set is then used to evaluate the accuracy and efficacy of the trained models.

Evaluation:
After initial training and testing, the pre-processed data set will be put to a comprehensive evaluation. This phase involves assessing the accuracy of the applied deep learning algorithms, such as CNN, RNN, and LSTM, in identifying and categorizing spam messages with accuracy. Utilizing evaluation metrics such as accuracy, precision, recall, and F1 score, the system's performance will be evaluated.

Retraining and Retesting:
In the final stage, a retraining and retesting phase is conducted with an unlabeled dataset and a deep learning algorithm. The unlabeled dataset is utilized in an unsupervised learning approach in which the deep learning algorithm learns data patterns and structures without explicit class designations. Using the information present in the unlabeled dataset, the model adapts and improves its ability to identify spam messages more accurately and efficiently through this iterative process.

4.1 Experimental Setup

4.1.1 Hardware

Deep learning heavily depends on computational capacity, and in the case of central processing units (CPUs), there are several crucial considerations. The CPU chosen for deep learning tasks can have a significant impact on performance and efficiency. CPUs are versatile and able to perform a variety of computational tasks, including deep learning. In our study, we used an Intel Celeron N3350 processor with 4 GB of memory. Central to the execution of deep learning algorithms and data processing was the CPU. CPUs may not provide the same level of parallelism as GPUs, but they excel at complex calculations such as convolution and backpropagation. CPUs provide a solid foundation for efficiently executing these repetitive computations.

4.1.2 Software

The architecture for machine learning use Anaconda Python 3.7 and TensorFlow as the primary framework for CPU-based computation. With its adaptability and effectiveness, TensorFlow provided a strong foundation for creating and setting up machine learning models. In addition, the TensorFlow-based Keras offered a user-friendly interface and support for various learning approaches. For data manipulation, analysis, pre-processing, and visualization, libraries including Pandas, NumPy, Scikit-learn, NLTK's Tokenizer, Seaborn, and Matplotlib were used.

5. RESULTS, ANALYSIS AND DISCUSSSIONS

Our proposed method considers essential training algorithm and parameter settings. Our training algorithm utilizes 10 epochs to effectively train the model. This number of epochs permits the model to acquire knowledge and improve its representations and predictions. Each epoch enables the model to adjust its parameters based on the training data, thereby identifying complex patterns within the dataset. In our experiments, the selection of 10 epochs strikes a balance between training duration and model performance, producing positive outcomes.

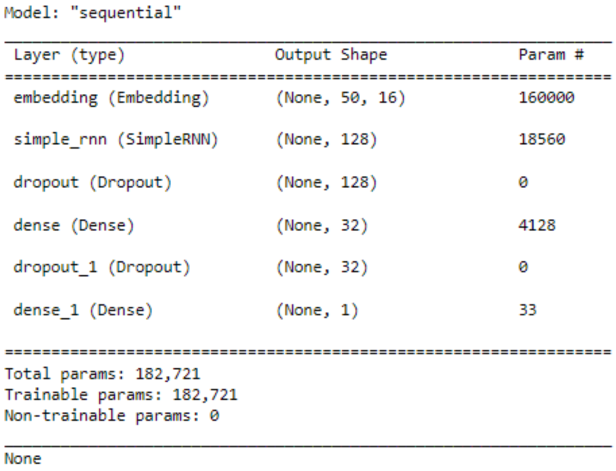
In the implementation, the RNN, CNN, and LSTM deep learning algorithms all use the same method. The efficacy of these algorithms in processing sequential data and extracting important features is well-known. Each word within a training vector has a fixed size of 50. The learning rate is set to adam optimizer, which is an adaptive learning rate optimization algorithm. By selecting 10 epochs, we ensure that the model has sufficient access to the training data, allowing it to improve on its predictions over time.

To evaluate the performance of our method on RNN, CNN, and LSTM, we conduct simulations on a same dataset. The dataset is divided into training and test sets, with test sizes ranging from 0.2 to 0.9. By training and testing the models with varying test sizes, we evaluate their robustness and generalization abilities across a range of unobserved data proportions. Using a test size of 0.2, we also measure performance metrics such as accuracy, precision, recall, and F1 score to evaluate the models' effectiveness in spam detection. The dataset is divided into training and test sets, with 80% (2627 records) for training and 20% (1035 records) for testing. These metrics provide insight into the ability of the models to correctly classify spam and non-spam messages, as well as their overall performance in terms of true positives, true negatives, false positives, and false negatives. By comparing the results of various test sizes, we obtain a comprehensive understanding of the performance of the models across a variety of scenarios and can make informed decisions regarding their suitability for deployment in the real world.

5.1 RNN

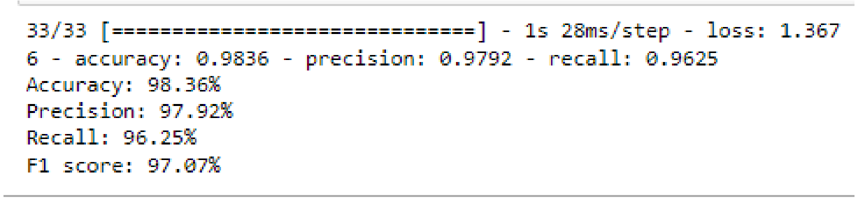
Recurrent Neural Networks (RNN) are a type of neural network that performs exceptionally well in sequential data processing tasks. RNNs have connections allowing information to pass through in a loop, allowing them to capture sequential dependencies. RNNs are particularly effective in applications such as natural language processing, speech recognition, and time series analysis. RNNs are well-suited for tasks involving sequential data and context due to their recurrent nature, which enables them to retain memory of previous inputs. Figure 3 shows the RNN Model Algorithm.

Figure 4 displays the RNN Model's most accurate result, which is 98.36%, in our proposed model. This result was obtained by dividing the dataset into 80% train and 20% test. Figure 5 compares the label distribution based on the accuracy demonstrated in Figure 4. In addition, Figure 6 illustrates the accuracy attained by the RNN Model for various test sizes. By altering the percentage of the dataset used for testing between 0.2% and 0.9%.



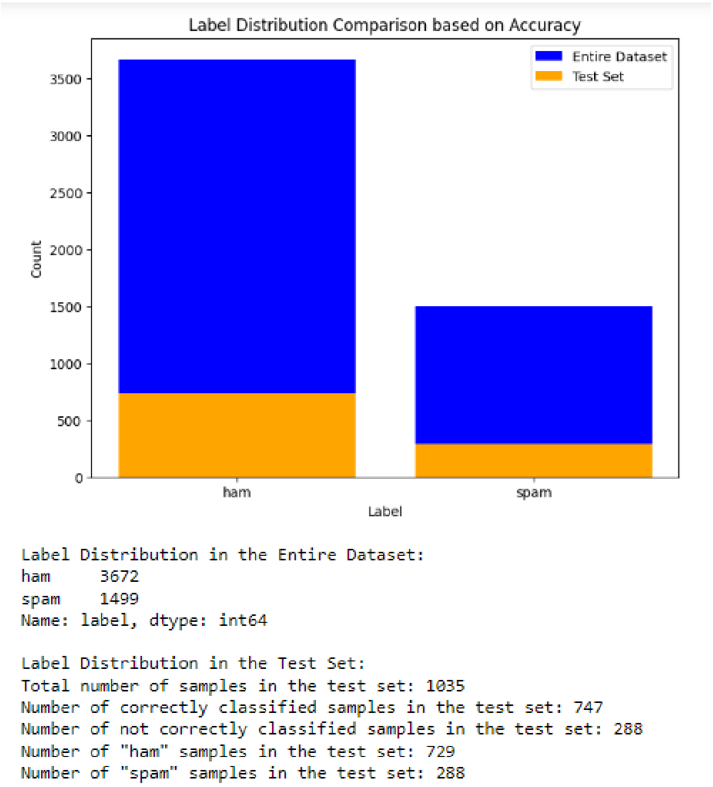
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 16)            160000

 simple_rnn (SimpleRNN)      (None, 128)               18560

 dropout (Dropout)           (None, 128)               0

 dense (Dense)               (None, 32)                4128

 dropout_1 (Dropout)         (None, 32)                0

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 182,721
Trainable params: 182,721
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3 RNN Algorithm Model



```
33/33 [==============================] - 1s 28ms/step - loss: 1.367
6 - accuracy: 0.9836 - precision: 0.9792 - recall: 0.9625
Accuracy: 98.36%
Precision: 97.92%
Recall: 96.25%
F1 score: 97.07%
```

Figure 4 Metrics of RNN Model



```
Label Distribution in the Entire Dataset:
ham     3672
spam    1499
Name: label, dtype: int64

Label Distribution in the Test Set:
Total number of samples in the test set: 1035
Number of correctly classified samples in the test set: 747
Number of not correctly classified samples in the test set: 288
Number of "ham" samples in the test set: 729
Number of "spam" samples in the test set: 288
```

Figure 5 Comparison of the label distribution

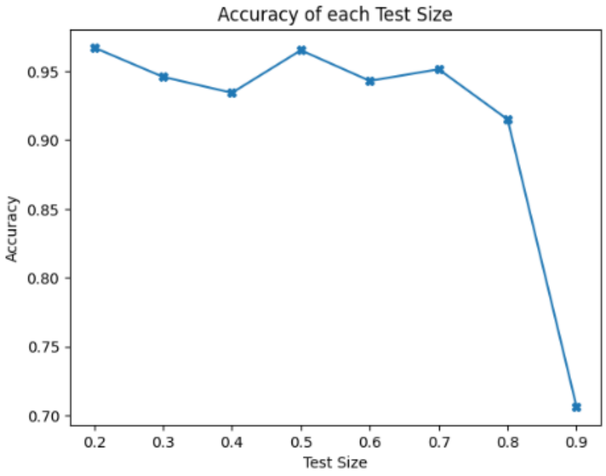
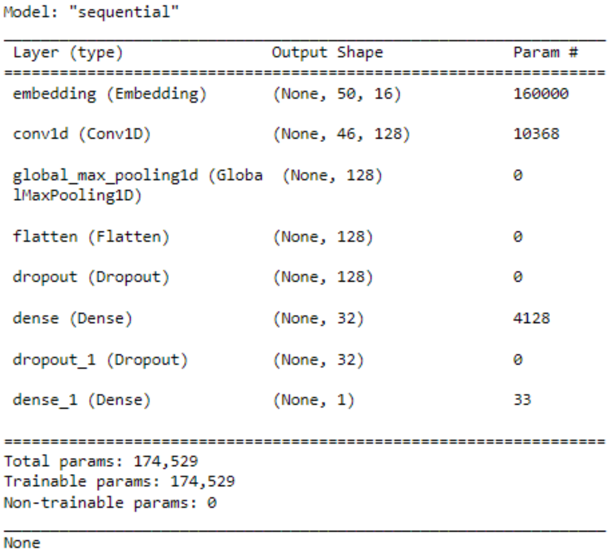


Figure 6 Different test size accuracy

5.2 CNN

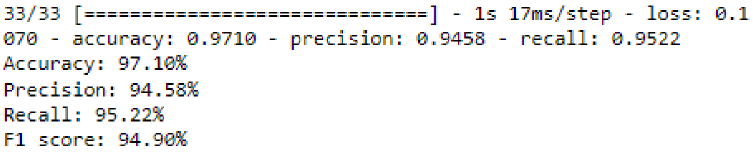
Figure 7 shows the CNN Model Algorithm. CNN is a deep learning model commonly used for image classification, object recognition, and text classification. CNNs are well-known for their outstanding performance in computer vision tasks, but they can also be applied effectively to text-based problems. CNNs can learn to identify relevant patterns and associations in sequential data, such as sentences and documents. CNNs use convolutional layers to extract local features and capture contextual information from text by treating it as a one-dimensional signal. These learned characteristics are subsequently passed on into layers with complete connectivity for classification.

Figure 8 displays the result of CNN Model result, which is 97.10%, in our proposed model. This result was obtained by dividing the dataset into 80% train and 20% test. Figure 9 compares the label distribution based on the accuracy demonstrated in Figure 8. In addition, Figure 10 illustrates the accuracy reached by the CNN Model across various test sizes. By lowering the proportion of the dataset used for testing between 0.2% and 0.9%.



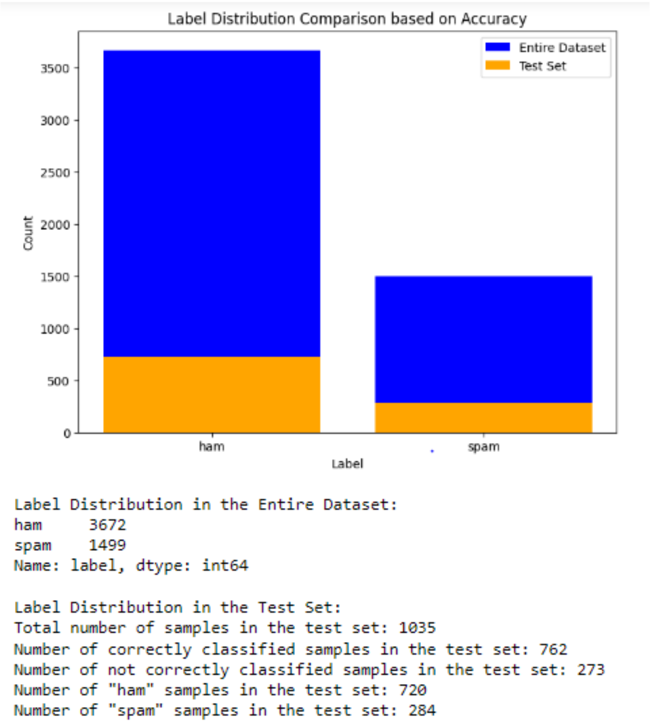
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding (Embedding)       (None, 50, 16)            160000    
                                                                 
 conv1d (Conv1D)             (None, 46, 128)           10368     
                                                                 
 global_max_pooling1d (Globa  (None, 128)              0         
 lMaxPooling1D)                                                  
                                                                 
 flatten (Flatten)           (None, 128)               0         
                                                                 
 dropout (Dropout)           (None, 128)               0         
                                                                 
 dense (Dense)               (None, 32)                4128      
                                                                 
 dropout_1 (Dropout)         (None, 32)                0         
                                                                 
 dense_1 (Dense)             (None, 1)                 33        
                                                                 
=================================================================
Total params: 174,529
Trainable params: 174,529
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 7 CNN Model Algorithm



```
33/33 [==============================] - 1s 17ms/step - loss: 0.1
070 - accuracy: 0.9710 - precision: 0.9458 - recall: 0.9522
Accuracy: 97.10%
Precision: 94.58%
Recall: 95.22%
F1 score: 94.90%
```

Figure 8 Metrics of CNN Model



```
Label Distribution in the Entire Dataset:
ham     3672
spam    1499
Name: label, dtype: int64

Label Distribution in the Test Set:
Total number of samples in the test set: 1035
Number of correctly classified samples in the test set: 762
Number of not correctly classified samples in the test set: 273
Number of "ham" samples in the test set: 720
Number of "spam" samples in the test set: 284
```

Figure 9 Comparison of the label distribution

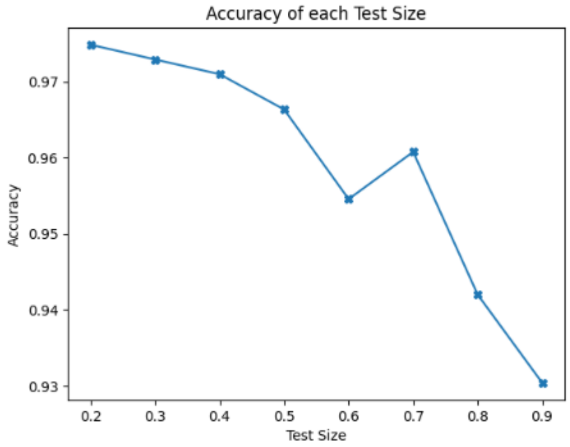


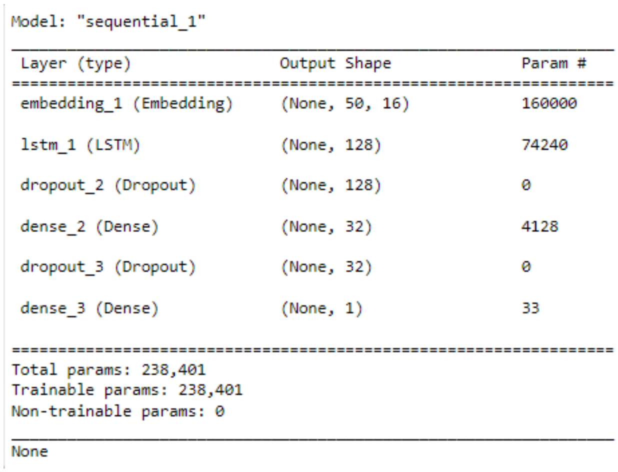
Figure 10 Different test size accuracy

5.3 LSTM

Long Short-Term Memory (LSTM) is a form of recurrent neural network (RNN) commonly used for processing sequential data, such as text classification. LSTM networks, unlike conventional RNNs, can encapsulate long-range dependencies and solve the vanishing gradient problem. They use a memory cell with gating mechanisms to selectively update and store data over time. In our proposed model, LSTM networks are used to comprehend the sequential nature of text and make precise predictions. Their capacity to capture both short-term and long-term dependencies makes them ideal for sentiment analysis, named entity recognition, and text generation. Figure 11 illustrates the LSTM Algorithm Model.

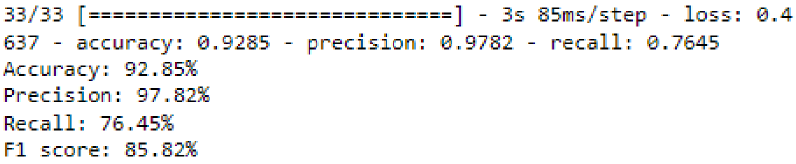
Figure 12 depicts the result obtained by the LSTM Model Algorithm in our proposed methodology, which achieved 92.85% accuracy. This result was achieved using a training set of 80% and a test set of 20%. Figure 13 depicts a comparison of the label distribution based on accuracy in Figure 12. In addition, Figure

14 illustrates the accuracy across various test sizes. These graphs illustrate the performance and efficacy of the LSTM Model described in our paper.



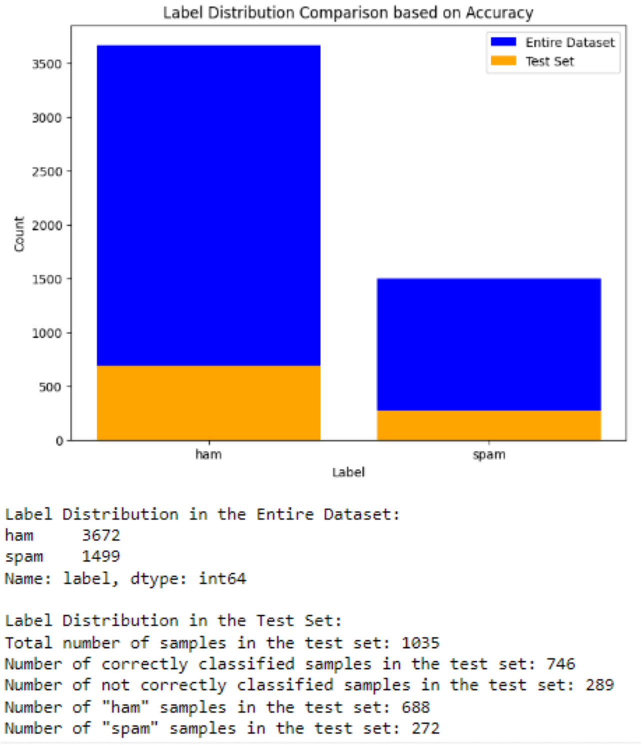
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 50, 16)            160000

 lstm_1 (LSTM)               (None, 128)               74240

 dropout_2 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 32)                4128

 dropout_3 (Dropout)         (None, 32)                0

 dense_3 (Dense)             (None, 1)                 33

=================================================================
Total params: 238,401
Trainable params: 238,401
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 11 LSTM Model Algorithm



```
33/33 [==============================] - 3s 85ms/step - loss: 0.4
637 - accuracy: 0.9285 - precision: 0.9782 - recall: 0.7645
Accuracy: 92.85%
Precision: 97.82%
Recall: 76.45%
F1 score: 85.82%
```

Figure 12 Metrics of LSTM Model



```
Label Distribution in the Entire Dataset:
ham     3672
spam    1499
Name: label, dtype: int64

Label Distribution in the Test Set:
Total number of samples in the test set: 1035
Number of correctly classified samples in the test set: 746
Number of not correctly classified samples in the test set: 289
Number of "ham" samples in the test set: 688
Number of "spam" samples in the test set: 272
```

Figure 13 Comparison of the label distribution

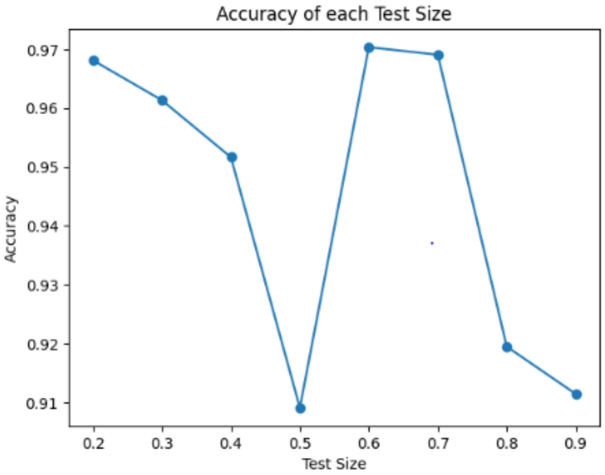


Figure 14 Different test size accuracy

6. CONCLUSION

In conclusion, our proposed model takes essential aspects of training deep learning algorithms and parameter settings with TensorFlow. We use 10 epochs to train the model effectively, allowing it to learn and refine its representations and predictions over multiple iterations. By using consistent methodologies for RNN, CNN, and LSTM, we can compare their performance under identical training conditions. Comparing these three algorithms revealed that the RNN obtained the highest accuracy of 98.36%, outperforming both the CNN and LSTM models. This demonstrates the RNN's efficacy in capturing the temporal dependencies present in sequential data and its capacity to extract meaningful patterns for spam detection. While CNN and LSTM also exhibited competitive performance, RNN demonstrated superior accuracy. These findings highlight the significance of selecting the optimal algorithm based on the specific task and dataset, as different algorithms may perform extremely well in different contexts. By simulating a specific dataset, we evaluate the robustness and generalization capabilities of the models across a range of test sizes. Metrics such as accuracy, precision, recall, and F1 score offer insight into the spam detection effectiveness of models. Overall, our experiments indicate that RNN, CNN, and LSTM are suitable for sequential data processing tasks, with each algorithm demonstrating exceptional accuracy and precision.

References

[1] Abuzaid, N. (2022). Image SPAM Detection Using ML and DL Techniques. International Journal of Advances in Soft Computing and Its Applications, 14(1), 227–243.

[2] Wu, T., Liu, S., Zhang, J., & Xiang, Y. (2017, January 30). Twitter spam detection based on deep learning. ACM International Conference Proceeding Series.

[3] Veda Reddy, T., Vinay Kumar, T., Laxmi Keerthi, T., & Johnson Joseph, C. (n.d.) (2020). International Journal of Research Email Spam Detection using Recurrent Neural Network.

[4] Chandra, A., & Khatri, S. K. (2019, November). Spam SMS filtering using recurrent neural network and long short term memory. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON) (pp. 118-122).

[5] Archchitha, K., & Charles, E. Y. A. (2019, September). Opinion spam detection in online reviews using neural networks. In 2019 19th International conference on advances in ICT for emerging regions (ICTer) (Vol. 250, pp. 1-6).

[6] AbdulNabi, I., & Yaseen, Q. (2021). Spam email detection using deep learning techniques. Procedia Computer Science, 184, 853–858.

[7] Douzi, Samira., AlShahwan, F. A., Lemoudden, Mouad., & Ouahidi, Bouabid. el.(2020). Hybrid Email Spam Detection Model Using Artificial Intelligence. International Journal of Machine Learning and Computing, 10(2), 316–322.

[8] Guo, Z., Shen, Y., Bashir, A. K., Imran, M., Kumar, N., Zhang, D., & Yu, K. (2021). Robust Spammer Detection Using Collaborative Neural Network in Internetof-Things Applications. IEEE Internet of Things Journal, 8(12), 9549–9558.

[9] Chakrabarti, S., Saha, H. N., University of British Columbia, Institute of Electrical and Electronics Engineers. Vancouver Section, Institute of Engineering & Management, University of Engineering & Management, & Institute of Electrical and Electronics Engineers. (n.d.). 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON): 17th-19th October 2019, University of British Columbia, Canada.

[10] Gowri, S. M., Sharang Ramana, G., Sree Ranjani, M., & Tharani, T. (2021). Detection of Telephony Spam and Scams using Recurrent Neural Network (RNN) Algorithm. 2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, 1284–1288.

[11] Najork, M. (2009). Web Spam Detection. Encyclopedia of Database Systems, 1, 3520-3523.

[12] Shafi'I, M. A., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A review on mobile SMS spam filtering techniques. IEEE Access, 5, 15650-15666.

[13] Britannica, T. Editors of Encyclopaedia (2021, February 26). spam. Encyclopedia Britannica. https://www.britannica.com/topic/spam

[14] Yegulalp, S. (2022, June 3). What is tensorflow? The Machine Learning Library explained. InfoWorld. Retrieved February 9, 2023, from https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learninglibrary-explained.html.

[15] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

[16] Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, 420 (2021). https://doi.org/10.1007/s42979-021-00815-1

[17] Yamashita, R., Nishio, M., Do, R. K. G., &amp; Togashi, K. (2018, June 22). Convolutional Neural Networks: An overview and application in radiology - insights into imaging. SpringerOpen. Retrieved February 9, 2023, from https://insightsimaging.springeropen.com.

[18] Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., &amp; Shah, T. (2022, February 3). Machine learning techniques for spam detection in email and IOT Platforms: Analysis and Research Challenges. Security and Communication Networks. Retrieved February 9, 2023, from https://www.hindawi.com/journals/scn/2022/1862888/.

[19] Dickson, B. (2020, November 30). How machine learning removes spam from your Inbox. Retrieved February 9, 2023

[20] Awati, R. (2022, September 29). What are convolutional neural networks? Retrieved February 9, 2023, from https://www.techtarget.com/searchenterpriseai/definition/convolutional-neuralnetwork.