2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020, Dhaka, Bangladesh

Email Spam Detection using Bidirectional Long Short Term Memory with Convolutional Neural Network

Sefat E Rahman
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna-9208, Bangladesh
Email: sefat.e.rahman@gmail.com

Shofi Ullah
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna-9208, Bangladesh
Email: mail2safiullah@gmail.com

***Abstract*—Communication over email in this era of Internet has become very popular on account of its being cheap and easy to use for messaging and sharing important information to others. But spam messages often times make large volume of unwanted messages in the users inbox and it also wastes the resources as well as valuable time of the users. Therefore, in order to identify the message whether it is spam or ham, an efficient and accurate technique is required. In this paper, we propose a new model for detecting spam messages based on the sentiment analysis of the textual data of the email body. We incorporate Word-Embeddings and Bidirectional LSTM network to analyze the sentimental and sequential properties of texts. Furthermore, we speed up the training time and extract higher level text features for Bi-LSTM network using Convolution Neural Network. We involve two datasets namely lingspam dataset and spam text message classification dataset and adopt recall, precision and f-score for comparing and evaluating the performance of our proposed approach. Our model achieves improved performance of accuracy about 98-99%. Apart from this, we demonstrate our model outperforms not only to some popular machine learning classifiers but also to state of the art approaches for detecting spam messages and hence, proves the superiority by itself.**

***Keywords*—** Email Spamming, CNN, Bidirectional LSTM, Word-Embeddings

I. Introduction

Almost all people prefer email system for their daily task to communicate with others. But spam message is a threat to both individual and society. Spam messages basically means that some undesired messages are sent to the special target users inbox. Spam messages make users inbox packed with some unwanted messages which not only annoy the user but also some times the target users may fall into the arranged trap of the attacker [1]. So, spam message is certainly a threat for email users as well as for the Internet society. Moreover, reading the whole texts of the spam message which is inboxed to the target users, some times user reads the whole text and afterward, realizes the message is spam and avoids it. Thereby, user wastes his valuable time too. That's why, it is very vital to prevent or filter out spam message from the messages which are not spam.

Most popular and widely used technique for spam detection is Machine Learning(ML) based algorithms. Some of the researchers conducted supervised ML algorithms for comparative performance for spam identification using like Naive Baiyes [2], Dicision Tree [3] SVM [4]. Integrated approach, combination of two or three algorithms are also proposed to improve the performance. Text based feature extraction is a very time-consuming task. Moreover, it is not easy to extract every important features from text which is not much in length. We demonstrate a new spam detection technique in this work using Bidirectional Long Short-Term Memory (Bi-LSTMs) with Convolutional Neural Network(CNN).

The *objective* of this research is to build a framework for email spam detection employing the neural network. We incorporate Convolutional Neural Network(CNN), Bidirectional Long-Short Term Memory(Bi-LSTM) with Word-Embedding Network. Our main contributions towards spam detection in this research are pictured as follows:

- We analyze the sequential and sentimental properties and relationships among texts of the email using Word-Embeddings and Bidirectional Long Short Term Memormy(Bi-LSTM).
- We speed up training time involving Convolutional Neural Network before the Bi-LSTM network and also extract higher level features of texts using this network within a less time compared to straight LSTM network.
- To evaluate the performance of the proposed approach, we employ lingspam and spam text message classification(STMC) datasets as well as computing precision, recall and f-score. Our model outperforms to some popular machine learning algorithm and also to some recent approaches used to detect spam messages.

978-1-7281-7366-5/20/$31.00

The remainder of this paper is embodied as follows; Section II describes literature study, Section III sketches overview and procedures of details implementation, Section IV presents the experimental results and finally, Section V ends with summarizing important outcomes of the experiment.

II. Literature Study

Various kinds of techniques and methods are used to detect spam email messages. There are actually three kinds of techniques used for this task namely single standard machine learning, hybrid machine learning, and feature engineering methods. Some works have also been done based on different kinds of features on textual and image data. Masurah [5] utilized Naïve Bayes, KNN, and Reverse DBSCAN algorithm by experimenting the Enron Corpus. For the recognition of texts, they adopted the OCR library and this OCR does not give so expected output.

As at the initial stage of pre-processing, image based spam can be filtered, text based email spam classification techniques were the focal point of most researchers. Some authors proposed to work based on clustering technique. Sasaki proposed k-means clustering [6] based approach to filter out spam from ham messages. Kumaresan [7] proposed spam detection technique, extracting textual features using SVM with cuckoo search algorithm. Renuka and Visalakshi utilized the SVM [4] to identify the spam email followed by the Latent Semantic Indexing (LSI) to select the features. TF-IDF [8] is used for the feature extraction. Here, SVM combined with LSI model compared with the SVM integrated with TF-IDF model without employing the LSI, PSO, and NN. However, SVM-LSI provides improved accuracy compared to any previous ML approach.

Feng proposed integrated SVM and NB approach [9] for filter out the spam email. Actually, Integrated approach develops overall accuracy compared to straight SVM and NB methods. Moreover, Negative Selection Algorithm(NSA) with Particle Swarm Optimization (PSO) algrorithm, proposed to classify spam email. Here, PSO is involved to optimize the performance of the classifier. In 2015, Idris utilized for the development of the Negative Selection Algorithm (NSA) with PSO [10] for spam email separation. But, PSO performed comparatively good with the random detector of NSA.

Tuteja and Bogiri [11] applied ANN based concept in 2016 to identify and filter the spam messages by creating the corpus manually. Also, K-means method was employed for extracting the features for this purpose. Zavvar proposed integrated approach [12] with Artificial Nueral Network(ANN), SVM and PSO. In 2019, Raj proposed LSTM based architecture [13] to filter out the spam messages and achieves good accuracy. However, all the methods stated does not perform so well. Moreover, accuracy is not good to that expected level in email spam classification task. Because in this particular case, if a single email can be considered as spam which is not spam actually, then it should be a matter of great concern to increase the accuracy and redesign the model again. That's why, we propose a new approach for spam message identification involving the text based features of the email body.

III. Working Methodology

Data preprocessing plays a significant role in natural language processing (NLP) as the real-world data are messy and contain unnecessary information and duplication. Major preprocessed steps are illustrated below.

Stop words removal: Stop words have very low or very high occurrence in the document but less significant in terms of importance. So, these are removed for better processing of the data.

Text Normalization: A word may have different order or lexicon form. So, in order to analyze properly, they all are needed to be converted to their root word. There are two techniques available for normalization namely stemming and lemmatization. Stemming just converts a word to its root by following some rules and it does not preserve context while conversion. On the contrary, lemmatization [14] is the combination of rule based and corpus based technique. Moreover, it preserves context of a word while converting to its root. That's why, we adopt lemmatization over stemming.

Word-Embeddings: In a word embedding method, words and documents are represented by a dense vector. Word-Embedding is the improvement over traditional bag-of-words model where huge sparse vectors were involved to utilize each word or to score each word within a vector to represent the whole vocabulary. As the vocabulary are huge and words or documents are represented with a large vector, therefore this representation is sparse.

On the contrary, in an embedding words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space. The position of a word within the vector space is taught from the text and focused on the words that surround the word when it is involved. In the learned vector space, position of a word is known as its embedding. We use pythonic keras embedding layer, which has 3 parameters:

- input length: It is basically the length of input sequences of the model. Input length is set to 500 for our proposed model.
- input dimension: This parameter refers to the size of the vocabulary in the texts. If the texts are integer encoded like values between 10-20, then vocabulary length would be 11. We encode our data as integer and set input dimension as 10000.
- output dimension: It defines the size of the output vectors from this layer for each word. We set output dimension to 40.

Now the resulting vector is passed as an input to the next stage of Convolutional Layer of the proposed architecture. The overall diagram of the proposed approach is pictured in figure 1.

Convolutional Network: Convolutional layer in Natural Language Processing has achieved surprising performance even for the textual data too. Generally to train the model containing

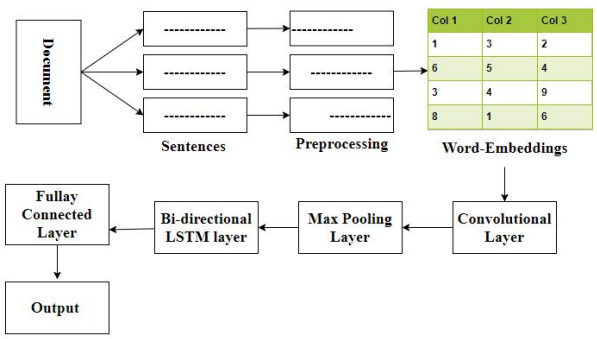

Fig. 1. Architecture of the Proposed Work

the textual sequential data, popular RNN model performs pretty but takes too much time. Using Convolutional layer before the RNN layer signifantly reduces the training time of the model. Moreover, higher level feature extraction [15] can also be possible through convolutional layer. Basically, the convolution layer targets to discover the combination between the different sentences or paragraphs of the document involving the filters. We adopt 64 one-dimensional features of size 5 of each. *Relu* activation function is used for this task. After that we employ the one-dimensional max pooling layer of pooling size 4 to extract the higher level features.

Bidirectional LSTM Network: For text sentiment analysis, Recurrent Neural Network(RNN) is very popular and widely used technique. The recurrent neural network (RNN) [16] remembers the previous time step information due to its having memory that facilitates it an advantage over traditional neural networks. The input of RNN is attached with a state vector to make new state vector. The resulting state vector remembers past information back from current. The straight forward RNN follows the following mathematical equations:

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{1}$$

$$y_t = W_{hy}h_t \tag{2}$$

Vanilla RNN [17] does not have capability of remembering the past sequence much. Apart from it, RNN has vanishing gradient descent problem. Long short-term memory network(LSTM) [18] is a variation of RNN, that is capable of learning long-term dependency and also succeed to solve vanishing gradient descent problem. LSTM is actually developed to resolve long-term dependency difficulty. LSTM has a special property of remembering. Main idea of the LSTM model lies simply in the *cell state*. The cell state flows straight through the sequence almost unchanged with only some little linear interaction. Another important thing about LSTM is *gate*. This gates control the information and information is safely added or deleted from the cell stated. LSTM model updates cell using the following equations:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3}$$

Here, x_t refers input, h_t refers the hidden state at t time step. The updated cell state C_t is as follows:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{4}$$

$$C_T = tanh(W_C[h_{t-1}, x_t] + b_C) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * C_T \tag{6}$$

Here, * is a point-wise multiplication operator and we can compute output and hidden state at t time step.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

LSTM faces difficulty as it only considers the previous contexts from current. So, both LSTM and RNN can only receive information from the previous time steps. So, to avoid this problem, further improvements are done using the Bidirectional Recurrent Neural Network (Bi-RNN). Bi-RNN [19] can handle two information both from the front and back. The combination of Bi-RNN and LSTM makes the Bi-LSTM. Therefore, the benefit of using LSTM is in the form of storage in cell state and Bi-RNN with access information from context before and after. Hence, it causes the Bi-LSTM to have the advantage of LSTM with feedback for the next layer. Bi-LSTM has additional another important advantage of remembering long term dependencies. The output will be based on call state and the output is regarded as the feature vector. Finally, the weighted sum of dense layer outputs is taken as an input of softmax activation function where we predict the probability of the email content as spam or ham.

We incorporate the three blocks namely Word-Embedding, Convolutional network and Bi-LSTM network, for separating the email messages based on sentiment and sequential properties of texts. Furthermore, we briefly illustrate below why these blocks are useful in email spam detection:

- We employ the Bidirectional LSTM network as the third block in the network on account of its reminding both previous and next sequence from the current. Moreover, it can understand and extract text sentiment and sequential characteristics more than the simple LSTM network.
- Secondly, we use the Convolutional Network block as the second block in the network before the Bi-LSTM block in order to extract the higher level and advanced features for the Bi-LSTM network. One of the reasons of adopting this block is to speed up the overall trainning time of the whole network as the Bi-LSTM takes much time to extract text based features.
- Convolutional block basically receives input as matrix form. Therefore, when converting texts to numeric matrix form, we prefer the word-embedding matrix representation which is not only convert words to its equivalent numeric form, but also takes semantic representation among words under consideration.

IV. Performance and Result Analysis

A. Dataset and Experimental Settings

We split the dataset as 80% for trainning and 20% for testing. Again from 80% trainning data, we split it to further 20% as for validation data for the model. We develop the entire project with python version 3.6. Moreover, We use NLTK pythonic library for the preprocessing task and for better processing of data, we use numpy and pandas library. Furthermore, we employ keras, tensorflow and scikit-learn pythonic library to build, compute and evaluate the performance of our proposed approach. We involve two datasets namely lingspam dataset and spam text message classification dataset. The description of datasets is given in brief in the table 1 below:

TABLE I
Descriptions of the Datasets

Dataset	Total Messages	Total Spam message(%)
Lingspam	2894	16.6
SPMDC	5573	14

B. Performance Evaluation

For analyzing the performance of our approach, precision, recall, and f-score are calculated considering the following parameters:

- True positive (TP) total no. of spam messages correctly detected as spam.
- False positive (FP) total no. of ham messages detected as spam.
- True negative (TN) total no. of ham correctly detected as ham.
- False negative (FN) total no. of spam messages detected as ham.

Now precision, and recall are computed by following equation

$$Recall, \alpha = \frac{TP}{TP + FN} \tag{9}$$

$$Precision, \beta = \frac{TP}{TP + FP} \tag{10}$$

F-score is the harmonic average of Precision and Recall and determines the test accuracy and determined as follow:

$$F\text{-}score = \frac{2 \times \alpha \times \beta}{\alpha + \beta} \tag{11}$$

Precision, recall and f-score metrics are computed for both datasets namely lingspam and spam text classification datasets. We also implemented some popular classifier like SVM, NB, RF, LSTM. In table II and III, we have shown the accuracy value in terms of these metrics.

From the table II and III, it is seen that our model achieves better performance score in terms of precision, recall and f-score metrics against with popular machine learning classifier namely Random Forest(RF), Naive Bayes(NB), Support Vector Machine(SVM) and vanilla LSTM. Furthermore, The proposed model accuracy and loss graph was recorded while we trainned the model. Fig. 2 and Fig. 4 shows the model accuracy grpah and Fig. 3 and Fig. 5 shows the loss graph respectively for both the datasets. In the following figure, it shows that the train and validation accuracy is being increased and also, train and validation loss is being reduced respectively with respect to increasing the number of epochs. The model obtained best accuracy after running it to 12 epocs when obtained accuracy about **98-99%** while minimizing the loss of the model.

TABLE II
Recall, precision, f-measure on liangspam dataset

Method	Recall (%)	Precision (%)	F-measure (%)
RF	86.67	86.87	86.87
NB	89.7	95	92.274
SVM	90	88.34	89.16
LSTM	96.78	95.5	96.13
Proposed	**98**	**98**	**98**

TABLE III
Recall, precision and f-measure on Spam Text Message Classification dataset

Method	Recall (%)	Precision (%)	F-measure (%)
RF	89	86.63	87.8
NB	87	92	89.43
SVM	89	87.59	88.3
LSTM	97	95.84	96.4
Proposed	**98.38**	**98.13**	**98.25**

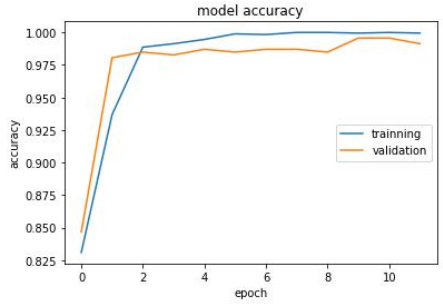

Fig. 2. Trainning and validation accuracy of the model for liangspam.

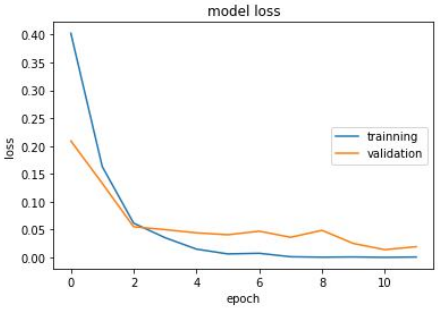

Fig. 3. Trainning and validation loss of the model for liangspam.

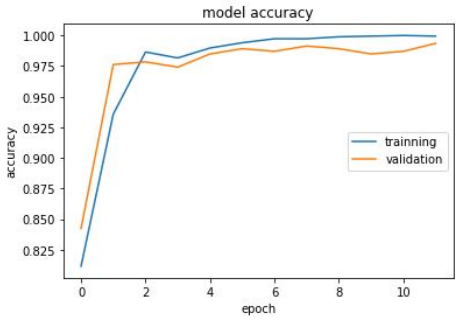

Fig. 4. Trainning and validation accuracy of the model for STMC.

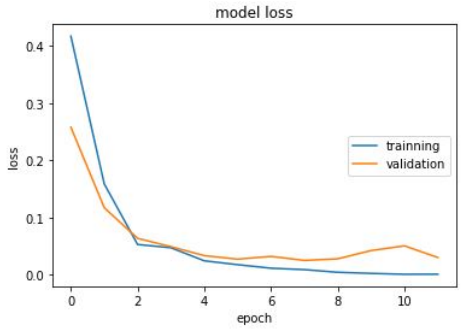

Fig. 5. Trainning and validation loss of the model for STMC dataset.

We also compared our proposed model performance with the some existing approach. The comparison is shown in the following table.

TABLE IV
ACCURACY COMPARISON WITH SOME EXISTING APPROACH

Method	Dataset	Accuracy(%)
SVM+LSI [4]	lingspam	93
S-cuckoo+SVM [7]	lingspam	89
LSTM [13]	lingspam	97
Naive Bayes+PSO [2]	lingspam	95.5
Proposed	lingspam/STMC	$>$**98**

From data in table 4, it is clearly seen that our proposed method simply outperforms than any other existing methods stated.

V. CONCLUSION

We present a new method in this research to detect an email message whether it is spam or ham focusing on textual analysis of email message. The model consists of three different networks namely Word-Embeddings, CNN and Bi-LSTM. We execute the trainning process of the model within less time using Convolutional layer after word-embedding layer and before LSTM network and also extract the higher level features for the Bidirectional LSTM network. To memorise the contextual meaning and the sequential property of a sentence, we adopt the Bidirectional LSTM network which makes the model very accurate giving improved performance accuracy about $98 - 99\%$.

REFERENCES

[1] R. Team, "Email statistics report, 2015-2019. the radicati group." 2019.
[2] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of naïve bayes and particle swarm optimization," in *2nd International Conference on Intelligent Computing and Control Systems(ICICCS)*. IEEE, 2018.
[3] S. Rajput and A. Arora, "Designing spam model-classification analysis using decision trees," *International Journal of Computer Applications*, vol. 75, no. 10, pp. 6–12, 2013.
[4] K. D. Renuka and P. Visalakshi, "Latent semantic indexing based svm model for email spam classification," 2014.
[5] M. Mohamad and A. Selamat, "An evaluation on the efficiency of hybrid feature selection in spam email classification," in *International Conference on Computer, Communications, and Control Technology (I4CT)*. IEEE, 2015, pp. 227–231.
[6] M. Sasaki and H. Shinnou, "Spam detection using text clustering," in *2005 International Conference on Cyberworlds (CW'05)*. IEEE, 2005.
[7] T. Kumaresan and C. Palanisamy, "E-mail spam classification using s-cuckoo search and support vector machine," *International Journal of Bio-Inspired Computation*, vol. 9, no. 3, pp. 142–156, 2017.
[8] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.
[9] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive bayes algorithm for spam filtering," in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. Las Vegas, NV, USA: IEEE, 2016, pp. 1–8.
[10] I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, "A combined negative selection algorithm–particle swarm optimization for an email spam detection system," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 33–44, 2015.
[11] H. Kaur and S. Ajay, "Improved email spam classification method using integrated particle swarm optimization and decision tree," *Next Generation Computing Technologies(NGCT)*, pp. 516–521, 2016.
[12] M. Zavvar, M. Rezaei, and S. Garavand, "Email spam detection using combination of particle swarm optimization and artificial neural network and support vector machine," *International Journal of Modern Education and Computer Science*, vol. 8, no. 7, p. 68, 2016.
[13] H. Raj, Y. Weihong, S. K. Banbhrani, and S. P. Dino, "Lstm based short message service (sms) modeling for spam classification," in *Proceedings of the 2018 International Conference on Machine Learning Technologies*, JINAN, China, 2018, pp. 76–80.
[14] K. Toutanova and C. Cherry, "A global model for joint lemmatization and part-of-speech prediction," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 486–494.
[15] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.
[16] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 234–239.
[17] S. Roy, S. I. Hossain, M. Akhand, and N. Siddique, "Sequence modeling for intelligent typing assistant with bangla and english keyboard," in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE, 2018, pp. 1–6.
[18] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.
[19] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.