Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 853–858

Procedia Computer Science

www.elsevier.com/locate/procedia

The 2nd International Workshop on Data-Driven Security (DDSW 2021)
March 23 - 26, 2021, Warsaw, Poland

Spam Email Detection Using Deep Learning Techniques

Isra'a AbdulNabi*, Qussai Yaseen

Department of Computer Information Systems, Jordan University of Science and Technology, 3030, Irbid 22110, Jordan

Abstract

Unsolicited emails such as phishing and spam emails cost businesses and individuals millions of dollars annually. Several models and techniques to automatically detect spam emails have been introduced and developed yet non showed 100% predicative accuracy. Among all proposed models both machine and deep learning algorithms achieved more success. Natural language processing (NLP) enhanced the models' accuracy. In this work, the effectiveness of word embedding in classifying spam emails is introduced. Pre-trained transformer model BERT (Bidirectional Encoder Representations from Transformers) is fine-tuned to execute the task of detecting spam emails from non-spam (HAM). BERT uses attention layers to take the context of the text into its perspective. Results are compared to a baseline DNN (deep neural network) model that contains a BiLSTM (bidirectional Long Short Term Memory) layer and two stacked Dense layers. In addition results are compared to a set of classic classifiers k-NN (k-nearest neighbors) and NB (Naive Bayes). Two open-source data sets are used, one to train the model and the other to test the persistence and robustness of the model against unseen data. The proposed approach attained the highest accuracy of 98.67% and 98.66% F1 score.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the Conference Program Chairs.

Keywords: Cybersecurity; Spam; BERT Transformer; Word embedding; Deep learning

1. Introduction

With the rising usage of the Internet and social networks, online communication has become an essential part of our daily life. One of the widely used mediums for formal and business communication is e-mail, because of its open access, quickness and reliability. With the email increasing popularity, spam emails were rapidly rising, spam is one or more unsolicited messages that take the form of advertising or promotional materials like debt reduction plans, getting rich quick schemes, online dating, and health-related products, etc. Automatic spam detection is no new topic [1]; business and companies are always looking to improve their user's experience. To protect their machines from potential damage; in case viruses were wrapped inside spam emails. Furthermore it helps saving network resources (bandwidth) and time from wastage. NLP and ML Communities have been attracted to resolve this problem and pro-

* Corresponding author. Tel.: +962-79-019-2509 ; fax: +962-2-709-5123.
E-mail address: ihabedulnabi18@cit.just.edu.jo

1877-0509 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2021.03.107

vided several SPAM detection data-sets and train models to classify whole document [2] [1]. Different classification approaches may perform worse or better, depending on the type of data and task they are given[3]. This has been achieved by employing the different NLP techniques; starting with the contribution of deep learning for understanding natural language. Convolutional neural network (CNN) that is used for sentence classification[4]. BiLSTM for sequence tagging[5]. Attention-based bidirectional LSTM networks for relational classification[6] and topic-based sentiment analysis[7]. Ending up with the state-of-the-art transfer learning models; where research focused on using pre-trained models for different tasks on NLP. BERT language model[8] showcase the highest results compared to all other works. BERT [9] is used to execute word embedding taking into account large left and right semantic contexts of words and can generate different semantic representations for the same word based on its context. In this research, BERT based Model is investigated for the task of automatic spam detection. The system is evaluated against different novel approaches on a publicly available corpus of spam and ham emails. The rest of the paper is organized as follows; Section 2 summarizes the related work. Section 3 describes the data and methodology used in this study. Section 4 shows and discusses the experiments results achieved and Section 5 concludes this work and what could be improved for future work.

2. Related Work

Spam e-mails detection problem has already drawn researchers' attention. Several significant works to detect spam e-mails have been proposed. In this section; prior related works that focus on the spam classification using ML and deep learning techniques are discussed.

Srinivasan et al. [10] present the effect of word embedding in deep learning for email spam detection, the proposed method performed better compared to other classical email representation methods.

Soni [11] proposed a profound learning model named THEMIS that uses an improved RCNN to recognize phishing messages showing the email header and the email body at both the character level and the word level. Test results gave the accuracy 99.84% for THEMES which is higher than both LSTM and CNN regarding their experiment.

Hassanpur et al. [12] represent emails to vectors using word2vec library instead of using rule-based methods. Vector representations are fed into a NN which is the learning model. Their approach achieves over 96% accuracy compared to the standard machine learning algorithms.

Egozi et al. [13] tried to approve the effectiveness of applying NLP techniques to detect phishing emails by processing the email samples content and extract features focused on word counts, stopword counts, punctuation counts, and uniqueness factors. The 26 extracted features were used to train an ensemble learning model based on linear kernel SVM and it was able to successfully identify over 80% of phishing emails and 95% of ham emails.

Seth et al. [14] propose a hybrid CNN model analyzing both the textual and image content of the email to classify it into spam or ham. Their model achieves a high accuracy of 98.87%.

Ezpeleta et al. [15] improve the accuracy of spam classification using Bayesian filtering classifiers up to 99.21% by adding polarity score feature which reflects the semantic of email content which concludes that sentiment analysis of the emails may help to detect spam emails.

Bibi et al. [16] propose a comparative study for previous spam filtering systems in terms of accuracy and dataset used. Table 1 presents a snip of it's review of recent works on spam detection systems that uses content base approach.

Table 1. Review of Recent Work on Spam Detection Systems.

Author	Classifier	Accuracy	Data Set
Awad et al [17]	NB, NN, SVM, KNN	99% achieved by Naïve Bayes	Spam assassin of 6000 instances
Saab et al. [18]	SVM, NB, LMSVM, Decision tree, ANN	93% achieved by SVM	Spam base of 4597 instances
Shajideen et al. [19]	SVM, NB, J48	94% achieved by SVM	3762 spam,5172 ham

3. Methodology

In this section, the overall approach and tools used to execute the spam email detection task are described in detail. Generally, any NLP task consists of five main phases: data collection, data pre-processing, feature extraction, model training, and model evaluation. Fig. 1 shows the flow for those phases. Hence, in this work, feature extraction will be done automatically as part of the deep learning model training,

Fig. 1. Genearl Flow of Main Phases for NLP Task

3.1. Data Collection

Tow open-source data sets were used in this work, both have two columns, the body text of the email and the class label spam or ham. The first data set is the open source Spambase data set from the UCI machine learning repository [20], the data set contains 5569 emails, of which 745 are spam. The second data set is the open source Spam filter data set from Kaggle [21] which contains 5728 emails of which 1368 are spam.
Exploring the distribution for SPAM and HAM classes, both data sets are imbalanced; where the SPAM class is the rare class. In order to avoid biasing for the major class which is the HAM class, a new balanced training data set is created through merging spam samples that are randomly selected from the two data sets Spambase and Spam filter with ensuring of no duplicate records. The new data set contains 2000 samples of SPAM and 3000 samples of HAM. In addition, hold-out portion from Spambase data set was reserved for testing the persistence of the model against unseen samples. The task is a binary-classification problem to detect whether the text is classified as SPAM 1 or HAM 0. Table 2 shows the distribution for both classes HAM and SPAM in both the training and testing data sets.

Table 2. Training vs Testing Data Samples.

Data set	SPAM	HAM	Total Samples
Dev data	2000	3000	5000
Hold out test data	113	113	226

3.2. Data Cleaning and Pre-Processing

After going through the class distribution for each target, the distribution of the word number in each record is explored. SPAM texts are longer than HAM texts in general, the input sequence for our model in this case study is defined to be 300 tokens or words. The next step was to clean the input data by extracting and removing the stop words using the Sklearn [22] library, since stop words provide little to no unique information that can be used for classification, then punctuation marks were extracted and removed as they affect the text encoding part, especially when it is attached with a word, for e.g: [users, user's] have different encoding. Text kept in cased shape as it might be an indicator for SPAM, especially promotion scams or other types of Spams. Keras tokenization tool [23] used to split the words as tokens based on the space. For example, ["We went to Amman."] it will be tokenized as the following: ['We', 'went', 'to', 'Amman.']. Each token got encoded to vector for classical classifiers using TfidfVectorizer from Keras. The label target variable is encoded to a binary format SPAM 1 and HAM 0.

3.3. Base Line Model

The baseline model is A state-of-the-art BiLSTM model [24]. It takes the input from an embedding layer and its output vectors are fed to a Dense layer. The Dense layer uses Relu activation function to allows converging quickly [25]. The next layer is a Dropout layer of 0.1 to avoid overfitting. Last layer is a Dense layer that uses Sigmoid activation function to normalize the output. Fig. 2 illustrate the baseline model structure and its layers.

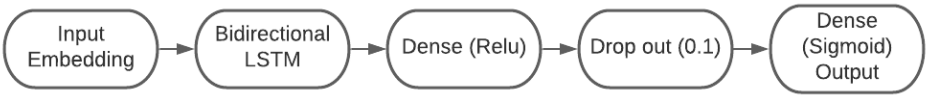

Fig. 2. Base-line DNN Model Structure layers

In addition to BiLSTM model, a set of classic classifiers KNN with n-neighbors=3 and MultinomialNB are trained to compare results with [26].

3.4. Transformer Model

BERT transformer is a pre-trained model published by Google AI language. It uses attention models to learn the contextual relation between the words in a sentence. It mainly consists of two parts: an encoder that encodes the input text and a decoder for the output result based on the task [27]. The bert-base-cased model implemented using Simple Transformers library [28] which is an API built above Hugging face library [29]. Bert cased model has been trained on English Wikipedia with 2500 million words and BookCorpus with 800 million words [30]. The model was trained using hyper-parameters: 300 sequence length, three epochs, 32 batch size, 4e-5 learning rate, and AdamW as an optimizer.

4. Experimental Results

The training data was split into 70% for training and 30% for validation. The cross-validation technique was applied since the data size is not very large to train the model, especially for the Neural network. Also, it gives better accuracy since n-folds contribute to the testing phase, n is set to eight. Two evaluation criteria have been used to compare the results Accuracy and F1 score. Please refer to Table 3 and Table 4 to observe the results for training data and the hold-out test data. In addition, Fig. 3 visualize the accuracy for training and hold-out data against all classifier in a bar chart.

Table 3. Testing Result on Training Data

Model	Accuracy	F1 Score
KNN	0.9310	0.9081
NB	0.9540	0.9408
BiLSTM	0.9650	0.9556
Bert Base Cased	**0.9730**	**0.9696**

From the result, as expected both DNN models achieve higher results than ML classifiers as their ability to learn the hidden features of the data improves their ability to classify. In comparison with the BiLSTM model that uses the Keras default word embedding; where each word is represented by a unique integer. Bert-base-case transformer model achieves the highest accuracy and F1-score. It profs that contextual representation for the input text does affect the spam classification result positively [31]. Finally, the models succeeded to persists against new test data as the

Table 4. Testing Result on Hold-out Data

Model	Accuracy	F1 Score
KNN	0.9292	0.9081
NB	0.9469	0.9459
BiLSTM	0.9643	0.9600
Bert Base Cased	**0.9867**	**0.9866**

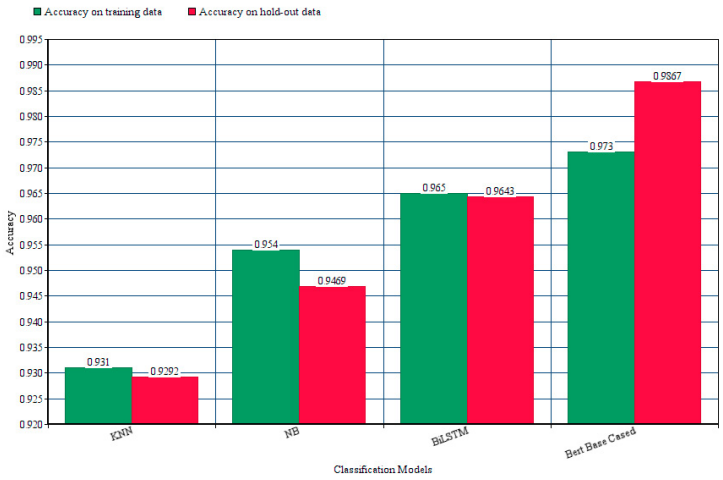

Fig. 3. Accuracy Comparison of Algorithms

accuracy remains almost the same if not better indicating that they are not over-fitted and the hyper-parameter fine-tuning process was a success also the data balancing and pre-processing.

5. Conclusion and Future Work

In this paper, state-of-the-art models were experiment against the task of detecting spam emails. Our results show that the bert-base-cased transformer model is the best model with an accuracy of 98.67% and 98.66% F1 score since it uses attention layers to take the context into its perspective. Bert contextual word embedding improves the capability of detecting spam emails compared to Keras word embedding that represents each word by a unique integer where it was applied in the BiLSTM model that achieves 96.43% accuracy and 96% F1 score. The results against unseen data reflect the persistence and robustness of the models that were perfectly fit. For future work, results can be improved even higher by taking a larger input sequence, the reason we stick with 300 sequence length is the limited GPU memory resource. Also, the SPAM detection task can be applied to another text language for e.g: Arabic.

References

[1] X.-L. Wang *et al.*, "Learning to classify email: A survey," in *2005 International conference on machine learning and cybernetics*, IEEE, vol. 9, 2005, pp. 5716–5719.

[2] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for spam filtering," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 2, p. 66, 2012.

[3] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[4] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[5] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 207–212.

[6] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 747–754.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[9] F. Del Vigna12, A. Cimino23, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on facebook," in *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, 2017, pp. 86–95.

[10] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A.-Z. Ala'M, and S. K. Padannayil, "Spam emails detection based on distributed word embedding with deep learning," in *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, Springer, 2021, pp. 161–189.

[11] A. N. Soni, "Spam-e-mail-detection-using-advanced-deep-convolution-neuralnetwork-algorithms," *JOURNAL FOR INNOVATIVE DEVELOPMENT IN PHARMACEUTICAL AND TECHNICAL SCIENCE*, vol. 2, no. 5, pp. 74–80, 2019.

[12] R. Hassanpour, E. Dogdu, R. Choupani, O. Goker, and N. Nazli, "Phishing e-mail detection by using deep learning algorithms," in *Proceedings of the ACMSE 2018 Conference*, 2018, pp. 1–1.

[13] G. Egozi and R. Verma, "Phishing email detection using robust nlp techniques," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2018, pp. 7–12.

[14] S. Seth and S. Biswas, "Multimodal spam classification using deep learning techniques," in *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, 2017, pp. 346–349.

[15] E. Ezpeleta, U. Zurutuza, and J. M. G. Hidalgo, "Does sentiment analysis help in bayesian spam filtering?" In *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2016, pp. 79–90.

[16] A. Bibi, R. Latif, S. Khalid, W. Ahmed, R. A. Shabir, and T. Shahryar, "Spam mail scanning using machine learning algorithm.," *JCP*, vol. 15, no. 2, pp. 73–84, 2020.

[17] W. Awad and S. ELseuofi, "Machine learning methods for spam e-mail classification," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 3, no. 1, pp. 173–184, 2011.

[18] S. A. Saab, N. Mitri, and M. Awad, "Ham or spam? a comparative study for some content-based classification algorithms for email filtering," in *MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference*, IEEE, 2014, pp. 339–343.

[19] N. M. Shajideen and V. Bindu, "Spam filtering: A comparison between different machine learning classifiers," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 2018, pp. 1919–1922.

[20] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[21] karthick veerakumar, *Spam filter*, 2017. [Online]. Available: https://www.kaggle.com/karthickveerakumar/spam-filter.

[22] C. Albon, *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. " O'Reilly Media, Inc.", 2018.

[23] N. Ketkar, "Introduction to keras," in *Deep learning with Python*, Springer, 2017, pp. 97–111.

[24] G. Liu and J. Guo, "Bidirectional lstm with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[25] J. Schmidt-Hieber *et al.*, "Nonparametric regression using deep neural networks with relu activation function," *Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, 2020.

[26] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019.

[27] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.

[28] T. Rajapakse. [Online]. Available: https://simpletransformers.ai/.

[29] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, arXiv–1910, 2019.

[30] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*, 2015. arXiv: 1506.06724 [cs.CV].

[31] G. Fan, C. Zhu, and W. Zhu, "Convolutional neural network with contextualized word embedding for text classification," in *2019 International Conference on Image and Video Processing, and Artificial Intelligence*, International Society for Optics and Photonics, vol. 11321, 2019, p. 1 132 126.