An Analysis of Machine Learning Algorithms and Deep Neural Networks for Email Spam Classification using Natural Language Processing

Type your text

Md. Mohidul Hasan
Computer Science(Software Engineering)
University of Hertfordshire
London, UK
shahan.hasan101294@gmail.com

Syed Mahbubuz Zaman
Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
syed.mahbub.uz.zaman@g.bracu.ac.bd

Md. Asif Talukdar
Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
md.asif.talukdar@g.bracu.ac.bd

Ayesha Siddika
Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
ayesha.siddika2@g.bracu.ac.bd

Md. Golam Rabiul Alam, PhD
Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
rabiul.alam@bracu.ac.bd

***Abstract*—Due to the extensive use of technology in our daily lives, email has become essential for online correspondence between individuals from all walks of life. As such certain individuals have weaponized this service by bulk mailing malicious emails to recipients with the goal of retrieving some form of classified information. Thus, Email classification has become a major area of research as it enables identification and isolation of such malicious emails. The objectives of this paper include a robust comparison of several traditional machine learning (ML) algorithms, exploring transfer learning with static (non-trainable) pretrained GLOVE (Global word vector representation) embedding, comparison of several deep learning models trained with GLOVE and keras embedding separately. Among ML classifiers, XGBoost achieved the highest evaluation scores. Among deep learning algorithms, keras embedding based models outperformed GLOVE embedding based models by a small margin which shows the efficiency of transfer learning in downstream NLP tasks (parts of speech tagging).**

***Index Terms*—XGBoost, Transfer Learning, Bi-directional Long Short Term Memory, Artificial Neural Network & Convolutional Neural Network.**

I. INTRODUCTION

Electronic mails (emails) play a significant role in day-to-day communication for a wide variety of professionals and businesses alike. Approximately A total of 319 billion emails are being sent and received per day in 2021 and this number is likely to grow over 376 billion by the end of 2025 according to email statistics report 2021 by RADICATI group [1]. As such malicious actors have begun using unsolicited emails to exploit users, customers or professionals of particular businesses. Despite the use of several spam email detection systems, the proportion of spam emails in total email traffic remains enormous [1], [2]. Statista states that a total of 45.1% of all emails exchanged in March, 2021 is identified as spam [3]. Traditionally, rules and protocol-based systems were employed to identify spam and phishing emails [4], [5]. These rule-based systems were static in nature rendering them ineffective against modern spam and phishing attempts [6]. Malicious attackers are growing more versatile in circumventing existing email filters as computational resources become more widely available. As such various machine learning based spam email detection systems have been proposed in the existing literature. The primary contributions of this paper include exploring transfer learning in training deep learning models (GLOVE embedding) as well as a comparison of the ML classifiers and Deep learning models using appropriate performance metrics.

II. LITERATURE REVIEW

A. Related Work

In their paper [7], I. AbdulNabi et al. trained a K-NN (K-nearest neighbour), NB (Naive bayes), Bi-LSTM (Bi directional Long short-term memory) and Google BERT model for email classification. These models were evaluated using Accuracy and f1-score. The results show that Bert out performed all the other models with an accuracy of 97.30% and an F1-score of 96.96%.

S. Srinivasan et al. in their paper [8], explored 3 Deep Convolutional Neural Network (DCNN) architectures as well as popular pretrained CNN architectures such as VGG29, Xception to classify spam images. The authors used several Image spam data-sets namely Image spam hunter data-set, an improved data-set developed by authors of [9] and Dredze ImageSpam data-set. These models were evaluated using accuracy and f1-score.

In their paper [10] S. Ishik et al. explored several Recurrent Neural Network architectures for email classification on Ag-

glutinative Language like Turkish. The data-set was collected from [11]. The authors trained an Artificial Neural Network (ANN), LSTM and Bi-LSTM with MI (Mutual Information) and WMI (Weighted Mutual Information) as feature selection. The authors show that Bi-LSTM received an accuracy of 100%.

Ankit Narendrakumar et al. in his paper [12], explored the efficacy of D-CNN algorithms for email classification. The authors used Enron and spam assassin data-sets to train the DCNN models. Finally, the author proposed THEMIS, an email classification model based on a mathematical approach where by the emails are divided into several sections and complex functions are employed to extract and classify the email signature. The models were evaluated using accuracy and f1-score. The proposed THEMIS model achieved an accuracy of 99.84%.

Alia. Barushka et al. in their paper [13], reviewed a spam classification models based on ANN, CNN, NB, SVM, Random Forest with ngram and skip gram word representation models respectively. The data-sets used by the authors include Cornell University positive hotel review spam and negative hotel review spam and TripAdvisor (Amazon Mechanical Turk). These models were evaluated using accuracy, AU-ROC, FN and FP. ANN and CNN models with a combination of ngram , skip-gram word representation out performed all the other models with an accuracy of 88.38% on the Negative data-set and 89.75% on the Positive data-set.

In their paper [14] Feng Wei et al. proposed a Bi-LSTM with GLOVE word Embedding to detect twitter bots. Cresci-2017 twitter data-set was used by the authors in their work. The model evaluation metrics include Precision, Recall, Specificity, Accuracy, F-Measure and MCC. The proposed model achieved an accuracy of 96.1%, a recall score of 97.6%, precision score of 94% and a specificity score of 93.5%.

Ismaila Idris in his paper [15], proposed an ANN with a negative selection algorithm (genetic algorithm) to classify spam and non-spam emails. The author contrasted the proposed model with an SVM classifier. The models were evaluated using train and test accuracy. The proposed model received a train accuracy score of 94.30% and a test accuracy score of 91.37%.

Sarit Chakraborty et al. in their paper [16] employed several variations of Decision tree classifier to filter spam emails from non-spam emails. The authors specifically used the NBTree Classifier, C 4.5 / J48 Decision Tree Algorithm and Logistic Model Tree Induction (LMT) classifier. These models were evaluated using accuracy. The authors show that LMT out performs all the other classifiers with an accuracy of over 85%, followed by NBTree with an accuracy of over 82% and lastly J48 with an accuracy of 78%.

Yoon Kim in his paper [17] employ variations of CNN model along with a pretrained word2vec word embedding to classify sentences. The CNN variations considered are CNN-rand, CNN-static, CNN-nonstatic and CNN-multichannel. The author concludes that simple CNN based architectures perform quite well in classification tasks related to Natural language using pretrained word2vec word representation model.

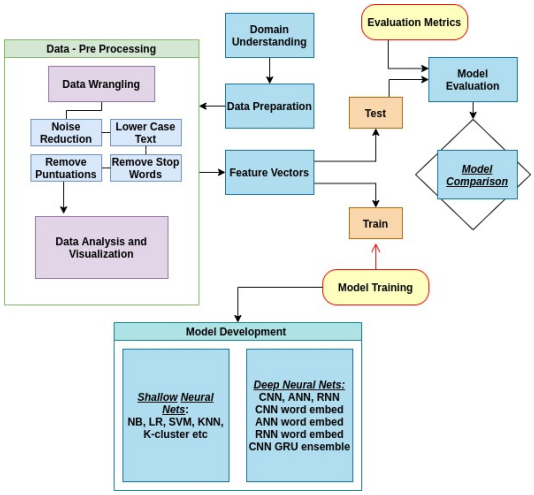

Fig. 1. Workflow diagram

B. Observation

Most research lack the use of appropriate performance metrics for model evaluation, as such it is difficult to conclude if these models are generalizing to the trained corpus or overfitting. The use of transfer learning and data wrangling in NLP is quite limited in existing literature. Our work incorporates these algorithms to provide a clear, concise and updated analysis of machine learning models in email classification.

III. METHODOLOGY

Supervised Classification tasks generally consist of six steps. These steps are defined as Data Acquisition and Preprocessing, Feature Extraction, Model Selection, Model Evaluation and Model Deployment. The figure 1 illustrates the steps performed within our work.

A. Data Acquisition and Pre-processing

The data set used in our work was acquired from the Enron data-set [18], a well-known publicly available benchmark corpus dedicated to spam email classification. Only the Kaminski folder of the data-set was used to generate the .csv file. We divided the dataset into a training set (80 percent) and test set (20 percent). Resulting training set had 4396 email samples and the test set had 1099 email samples.

Several data wrangling/pre-processing steps were performed on raw emails towards optimising the data-set for the purpose of spam email classification. These steps include **Normalization** (removing repeating emails, stopwords, words less than 3 words and punctuation), **Tokenization** and **Lemmatization**. Stopwords and puntuations usually hold negligible value when it comes to classification of texts or documents but they may be very useful in predicting words, completing sentences and other similar tasks.

Usually raw texts contain empty spaces, new line characters or other document specific symbols. For the purpose of our

task the raw emails were converted into a list of words where each word is referred to as a token (Tokenization). Words in a document are used in varying forms due to grammatical requirements. For example : **Democracy - Democratic - Democratization**. Lemmatization converts these words to their base, root or dictionary form. This allows optimal feature extraction as words used in varying contexts will have the same base form. **These tokens were lemmatized and converted back to sentences/emails.**

B. Feature Extraction

Machines are unable to process natural language such as text in English. For this reason linguistic data is required to be transformed into a numeric representation which concisely encapsulates the statistical inferences (Distribution, Frequencies) of the data as well as contextual and semantic meaning in many cases. This numeric representation is used as features for training ML models.

1) ***TF-IDF (Term Frequency - Inverse Document Frequency)****:* TF-IDF was used to train traditional ML classifiers (Not neural network based) within our work. TF-IDF score is assigned to a word based on the frequency of the word in a document and the number of documents it exists in. Generally, within linguistic data or corpus certain words are used more frequently despite retaining lower significance or relevance contrast to certain other words used rarely despite holding higher relevance to the meaning of the message. TF-IDF score is used to balance out the weights assigned to words such that frequent non relevant words hold lower values compared to infrequent highly relevant words. That is, the TF-IDF score gives more meaning to rare terms in the corpus and penalizes more commonly occurring terms [19].

2) ***Word Embedding****:* Word Embeddings are able to represent linguistics items or words in a low dimensional vector space. These numeric vectors (of words) are grouped together within the vector space/ word space based on semantic similarity. for instance, Boat - Ship. There are primarily two ways to train word embeddings, namely: Learnable Embedding (Keras Embedding), Pre trained word embedding (GLOVE). The DNN models within this work were trained using keras and GLOVE embedding separately with varying architectures.

a) ***Keras Embedding*** *:* Keras Embedding requires a specific input and output dimensions as arguments. The input texts/words are required to be converted into a one hot encoded vector prior to training the Embedding matrix. The parameters of the Keras Embedding matrix is updated during Gradient Descent. During our work a 100-dimension word embedding was trained using Keras Embedding layer, meaning that, each word from the corpus/emails was transformed to a 100-dimensional vector.

b) ***GLOVE Embedding****:* GLOVE is a pre-trained word embedding model developed by [20]. GLOVE employs both global statistics of matrix factorization like LSA(Latent Semantic Analysis) and word2vec model. Pennington has published several GLOVE embedding matrices of varying dimensions (50,100,200,300). For this study we have employed the 100-dimensional GLOVE embedding matrix. The GLOVE Embedding layer was **static** that is, GLOVE embedding matrix was **not fine-tuned (not updated during Gradient Descent)** towards email classification.

C. Model Selection

1) Machine Learning Classifiers (ML): The traditional machine learning classifiers trained within this work include, *Multinomial Naive Bayes*, *Random Forest*, *Decision Tree*, *Gradient Boosting*, *XGBoost*, *Logistic Regression*, *K-nearest neighbors*, *SVM* and *SVM(RBF)*.

2) Deep Neural Network Classifiers (DNN):

a) ANN – Artificial Neural Network: ANN is a feed forward neural network that can identify patterns within data. ANN comprises of several interconnected layers of nodes. The connections between the nodes have adjustable parameters. These parameters, along with the connections among the nodes, determine the output of the ANN.

b) Bi-LSTM – Bi-Directional Long Short-Term Memory: RNN (Recurrent Neural Network) specializes in processing sequential or time-dependent data because of their ability to utilize context (retained memory across inputs) when making final predictions. Bi-LSTM is a type of RNN that process sequential data in forward (past) and backward (future) directions making them more efficient in sequential learning tasks (Machine translation).

c) CNN – Convolutional Neural Network: CNN's employ convolution operations on the data matrix to reduce its dimensions while retaining important features of the dataset. CNN's take in sequential data (text) as a 1-dimensional matrix and, consequently, perform 1-dimensional convolution operation.

D. Model Evaluation

The classifiers and neural network models trained within this work were evaluated using the following metrics.

*True Positives(TP):*Total number of spam emails correctly recognized.

*True Negatives(TN):*Total number of benign/ham emails correctly recognized.

*False Positives(FP):*Total number of ham emails falsely recognized as spam emails.

False Negatives (FN): Total number of spam emails falsely recognized as ham emails.

Precision: Precision in this works context is defined as the ratio of predicted spam emails and true spam emails.

$$tp/(tp + fp) \tag{1}$$

High precision means the model predicts low false positives and high true positives.

Recall: Recall in this works context is defined as the ratio of true spam emails and predicted spam emails.

$$tp/(tp + fn) \tag{2}$$

The higher the Recall, the higher the model identifies the positive events and labels correctly. F1 score: F1 score is the weighted average of precision and recall

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \quad (3)$$

The best performing models have an F1 score close to 1. *Accuracy:* Accuracy is the ratio of correctly classified emails (both spam and ham) among all emails in the test or train set.

$$(tp + tn)/(tp + fp + fn + tn) \quad (4)$$

AU-ROC: Receiver Operating Characteristics (ROC) is a probability distribution curve for both tp and tn. Area under curve (AUC) of ROC is the measure of separation that is the ability of a model to distinguish between classes correctly.

IV. RESULT ANALYSIS AND DISCUSSION

A. Experimental Setup:

The traditional machine learning models were trained on a laptop using a jupyter notebook environment. The Deep learning models (ANN, CNN, Bi-LSTM) were trained using google colab GPU and high ram configuration. Libraries used within this work include: Pandas, Numpy, Seaborn, Matplotlib, WordCloud, Scikit-learn, Keras, NLTK and Tensorflow.

B. Comparison of traditional machine learning classifiers:

Table I, illustrates a comparison of all ML classifiers trained in our work. The table shows that XGBoost achieved the best scores for recall, f1 score, accuracy and AU-ROC. SVM (RBF) achieved the best precision score. SVM (Linear) achieved the second-best evaluation scores. KNN and Decision tree achieved the lowest evaluation scores. All the classifiers have received evaluation scores of over 95 percentile which was expected and an improvement over [4], [6], [7], [16], [21].

[16] shows that word embedding with CNN-LSTM achieves an accuracy of 95.9 %, recall of 1.0, precision of 0.936, f1 score of 0.967 and a G-mean of 96.7 %. This paper also shows FastText email representation in conjunction with CNN-LSTM achieves the same evaluation scores as word embedding with the exception of precision which is 93.5%. [6] shows that Text CNN achieves an accuracy of 97.54 % and f1 score of 0.97. [7] shows a Bert based model (Best performing model) with accuracy of 0.9730 and f1 score of 0.9696 on the training set and that of 0.9867 and 0.9866 respectively on the holdout set. [16] shows a Logistic model tree classifier with an accuracy of 85.9%. [21] shows that their proposed model QUAGGA produces a precision, recall and accuracy score of 0.98 respectively.

The top 3 classifiers (XGBoost, SVM-Linear, SVM-RBF) within our work outperform all the models from the [4], [6], [7], [16], [21].

C. Comparison of Deep Neural Network (DNN) models

Table II, illustrates a comparison of all Deep learning models (ANN, CNN, BI-LSTM) with keras embedding and pretrained GLOVE embedding trained in our work. The table shows that keras Embedding based DNN models outperform pretrained GLOVE embedding based models by a very small margin in terms of evaluation scores.

The GLOVE based DNN models were static in nature which means it was used in conjunction with the DNN models and were not updated or fine-tuned during Gradient Descent (training). As such these models had a significantly lower number of trainable parameters compared to keras embedding based models. Despite being static in nature, GLOVE based models achieved very high evaluation scores. The reason being pre-trained word embedding models (word2vec, GLOVE) encapsulate word similarities off the shelf.

Among GLOVE based models, ANN GLOVE-1 and ANN GLOVE-2 have the lowest evaluation metrics which was expected due to complexity of the problem, structure of the data and the general working principle of artificial neural networks (ANN).

Overall, the DNN models trained in this work outperform all other models proposed within the literature specifically [4], [6], [7], [16], [21].

Figure 2, shows heat-maps of classification report for DNN models (both keras and GLOVE embedding based). GLOVE based ANN models have the highest while keras embedding based models have the lowest false positives and false negatives respectively.

Figure 3, shows the AU-ROC curves for DNN models. ANN GLOVE-1 and ANN GLOVE-2 incurred the lowest AU-ROC scores because of low precision and recall as well as high false negatives and false positives. All other DNN models have an AU-ROC score of over 0.95 which means these models have generalized to the imbalanced data-set and were able to distinguish spam and ham emails with moderately high accuracy.

V. CONCLUSION

The primary objective of this paper was threefold. We have provided a concise comparison of traditional machine learning algorithms (Naive Bayes, SVM) for email classification using Enron corpus. XGBoost achieved the highest evaluation scores among other classifiers. We have trained six DNN models using pre-trained GLOVE embedding and three DNN models using keras embedding. We have provided a rigorous comparison of these nine DNN models. Keras embedding based DNN models due to their large number of trainable parameters (Table II) have outperformed other models and classifiers. We have also observed that pretrained GLOVE embedding based DNN models (CNN GLOVE-1, CNN GLOVE-2, Bi-LSTM GLOVE-1 and Bi-LSTM GLOVE-2) have achieved extremely high evaluation scores. This shows that transfer learning can be extremely useful and, in many cases, better for downstream NLP tasks like text classification.

Some future works include, using all Enron directory to generate a balanced data-set with 0.5 million messages or emails approximately. Generating Adversarial Attacks to evaluate the robustness of the trained models, implementing Google Bert embedding layer and using Bert models.

Fig. 2. Heat-map of the classification report for Deep learning models

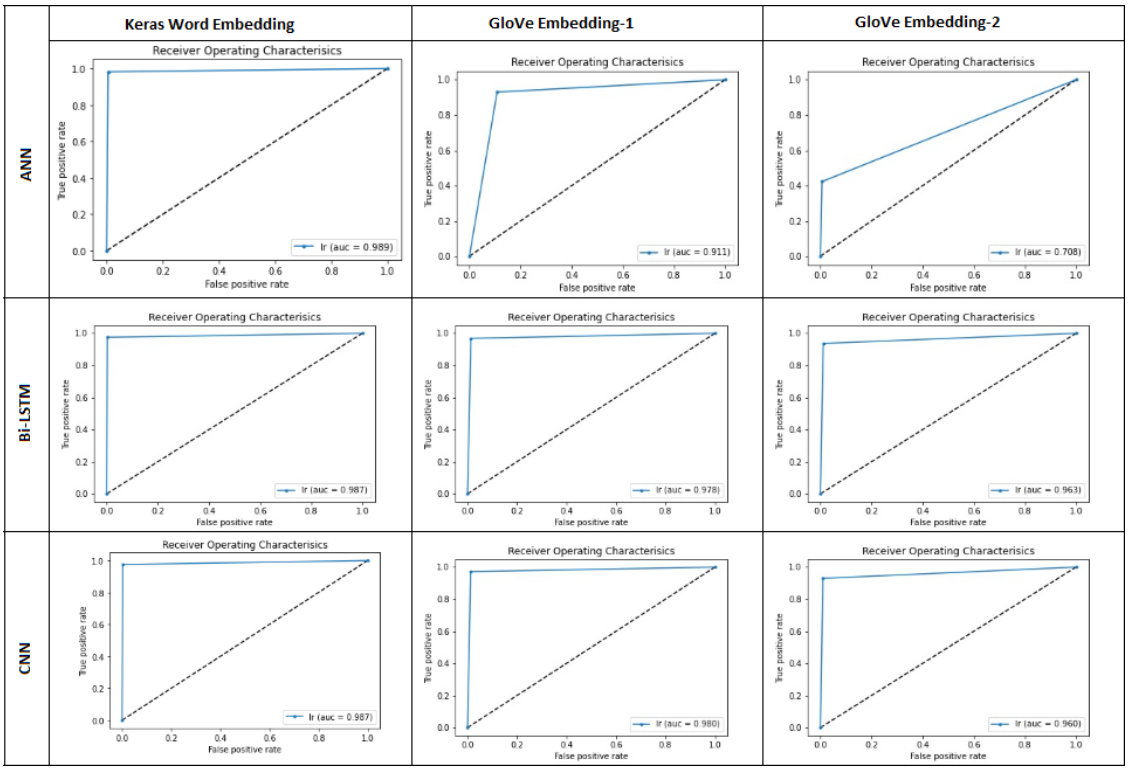

Fig. 3. AU-ROC for DNN-Classifiers

TABLE I
COMPARISON OF MACHINE LEARNING CLASSIFIERS

Model	Precision	Recall	f1 score	Accuracy	AU-ROC	Train Time(s)
XGBoost	0.9844	**0.9898**	**0.9871**	**0.9900**	**0.9898**	18.89
SVM (Linear)	**0.9868**	0.9801	0.9833	0.9873	0.9801	0.15
SVM (RBF)	0.9879	0.9789	0.9833	0.9873	0.9789	15.33
Logistic Regr.	0.9861	0.9644	0.9746	0.9809	0.9644	2.16
Gradient Boosting	0.9787	0.9620	0.9699	0.9773	0.9620	24.85
Random Forest	0.9816	0.9458	0.9620	0.9718	0.9458	7.06
Decision Tree	0.9602	0.9419	0.9506	0.9627	0.9419	2.92
KNN	0.9625	0.9350	0.9477	0.9609	0.9350	0.31
MultinomialNB	0.9352	0.7885	0.8312	0.8899	0.7885	0.02

TABLE II
COMPARISON OF DEEP LEARNING MODELS

Model	Precision	Recall	f1 score	Train Accuracy(%)	Test Accuracy(%)	Error	Loss	AU-ROC	Train Time(s)
ANN(Keras embedding)	**0.9888**	**0.9894**	0.9899	99.91	**99.18**	0.81	0.02	**0.9888**	48.53
CNN(Keras embedding)	0.9865	0.9893	**0.9922**	100	**99.18**	0.81	0.03	0.9865	54.47
Bi-LSTM(Keras embedding)	0.9789	0.9833	0.9879	99.82	98.73	1.27	0.05	0.9789	119.11
CNN(GLOVE embedding-1)	0.9799	0.9788	0.9777	100	98.36	1.63	0.69	0.9799	942.53
Bi-LSTM(GLOVE embedding-1)	0.9775	0.9764	0.9754	97.52	98.18	1.81	0.07	0.9775	3592.29
Bi-LSTM(GLOVE embedding-2)	0.9630	0.9678	0.9728	98.27	97.54	2.45	0.05	0.9630	4142.67
CNN(GLOVE embedding-2)	0.9601	0.9665	0.9733	99.36	97.45	2.54	0.13	0.9601	202.71
ANN(GLOVE embedding-1)	0.9109	0.8810	0.8623	88.81	90.17	9.82	0.25	0.9109	806.91
ANN(GLOVE embedding-2)	0.7085	0.7461	0.8954	81.30	84.53	15.46	0.34	0.7085	318.35

REFERENCES

[1] "Email statistics report." [Online]. Available: https://www.radicati.com/wp/wp-content/uploads/2021/EmailStatisticsReport,2021-2025ExecutiveSummary.pdf

[2] J. Johnson, "Spam statistics: Spam e-mail traffic share 2019," Jul 2021. [Online]. Available: https://www.statista.com/statistics/420391/spam-email-traffic-share/

[3] P. by Statista Research Department and O. 21, "Global average daily spam volume 2021," Oct 2021. [Online]. Available: https://www.statista.com/statistics/1270424/daily-spam-volume-global/

[4] 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.

[5] S. Nazirova, "Survey on spam filtering techniques," Aug 2011. [Online]. Available: https://www.scirp.org/journal/paperinformation.aspx?paperid=6769

[6] 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.

[7] Q. Yaseen *et al.*, "Spam email detection using deep learning techniques," *Procedia Computer Science*, vol. 184, pp. 853–858, 2021.

[8] S. Srinivasan, V. Ravi, V. Sowmya, M. Krichen, D. B. Noureddine, S. Anivilla, and K. Soman, "Deep convolutional neural network based image spam classification," in *2020 6th Conference on data science and machine learning applications (CDMA)*. IEEE, 2020, pp. 112–117.

[9] A. Chavda, K. Potika, F. D. Troia, and M. Stamp, "Support vector machines for image spam analysis," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: BASS,*, INSTICC. SciTePress, 2018, pp. 431–441.

[10] S. Isik, Z. Kurt, Y. Anagun, and K. Ozkan, "Spam e-mail classification recurrent neural networks for spam e-mail classification on an agglutinative language," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 4, pp. 221–227, 2020.

[11] L. Özgür, T. Güngör, and F. S. Gürgen, "Adaptive anti-spam filtering for agglutinative languages: a special case for turkish," *Pattern Recognit. Lett.*, vol. 25, pp. 1819–1831, 2004.

[12] 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.

[13] A. Barushka and P. Hajek, "Review spam detection using word embeddings and deep neural networks," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2019, pp. 340–350.

[14] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2019, pp. 101–109.

[15] I. Idris, "E-mail spam classification with artificial neural network and negative selection algorithm," *International Journal of Computer Science & Communication Networks*, vol. 1, no. 3, pp. 227–231, 2011.

[16] S. Chakraborty and B. Mondal, "Spam mail filtering technique using different decision tree classifiers through data mining approach-a comparative performance analysis," *International Journal of Computer Applications*, vol. 47, no. 16, 2012.

[17] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[18] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*. Springer, 2004, pp. 217–226.

[19] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for idf," *Journal of Documentation - J DOC*, vol. 60, pp. 503–520, 10 2004.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[21] T. Repke and R. Krestel, "Bringing back structure to free text email conversations with recurrent neural networks," in *European Conference on Information Retrieval*. Springer, 2018, pp. 114–126.