Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Improved Email Spam Detection Using An Integrated Approach Of Lstm And Attention Mechanism

Dr. GURURAJ A NAGALIKAR

Assistant professor Department of Computer Science,
Government First Grade College Shorapur, Dt. Yadgir

Abstract: When it comes to sharing information over the internet, email is a must-have. Spamming has emerged as a profitable industry thanks to the proliferation of email. Spam refers to unsolicited electronic mail or text communications sent to Internet users. Spamming entails distributing commercial communications to large numbers of people who have not requested them. These unwanted communications not only clog up the network's resources (especially its memory) but may also be exploited in attacks. The target of such an assault might have his data or identity exposed to others, or both. In this study, we explore an LSTM (long short term memory) integrated approach to spam detection.

Keywords: Long Short Term Memory LSTM, Email, spam detection, web data communication

Introduction

About 168 million emails are sent per minute all across the globe. What used to be our best friend is now holding us back and causing unnecessary anxiety: 87% of workers experience email overload, with 53% reporting they are overwhelmed by the constant stream of messages [1]. People struggle to prioritize the constant stream of incoming emails, sometimes needing to scroll through five to ten messages before locating one that really needs their immediate attention. Actually, some emails are better than others [2]:

- 15% of each category requires action. Even within emails, some are more urgent than others and must be attended to right once, while others may wait. The project's objective is to alleviate people's stress caused by excessive email and free up time that may be put to better use. Our approach relies heavily on deep-learning algorithms for effective categorization, prioritization, and presentation of email content. We broke down email handling into three distinct phases::

1. Reception:

i. Assign a priority to each incoming email, allowing users to prioritize some messages over others.
ii. Emails should be filed according to the headings in the above list..

The internet has become more crucial to modern society. The number of people who use email steadily rises in tandem with the growth of the internet. Every day, over 294 billion emails are sent. Unwanted, mass email transmissions (Spam) are a growing concern as email use rises [1]. About 90% of all emails received daily are thought to be spam or malware. Spam emails are a byproduct of email's rise to prominence as a marketing tool. Emails that the recipient has not requested, or "spam," are unwanted. Emails with almost similar content are sent to many different people[3]. The ever-increasing flood of spam emails is wreaking havoc on ISPs, end users, and the whole Internet infrastructure. Denial of service, in which spammers overwhelm an email server with traffic, is an example of this. This may slow down the delivery of valid messages. In addition to being a waste of time and effort, bandwidth, storage space, and processing power, spam emails may also be malicious. In addition, recipients lose time and effort sifting through their inboxes for real messages amid the junk and then taking steps to delete the spam. Spam is tough to deal with and categorize[4].

Furthermore, new spams are always emerging, and these spams are frequently purposefully customized so that they are not discovered, further hindering accurate detection, and a single model cannot solve the issue.

A spam filter is a piece of software designed to identify and block unwanted electronic mail before it reaches its intended recipient. A spam filter, like other filtering systems, uses its own set of criteria to determine whether or not an email is spam. The oldest and most basic versions (like the one included with Microsoft's Hotmail) allow the user to filter out messages based on keywords in the subject line. Due to the possibility of missing valid communications (known as false positives) and forwarding real spam messages, this approach is not very successful.Bayesian filters and other heuristic filters, which are used by more complex systems, aim to detect spam by suspicious word patterns or word frequency.

To this end, several studies have developed and evaluated different machine learning (ML) methods and models. Several comparative studies [5, 6, 7] aimed to evaluate several models for their ability to spot spam in email. Maximum accuracy for the random forest classifier [7] is 94.2%. To boost their effectiveness, several studies [8, 9]-[10] have combined ML algorithms with bio-inspired algorithms such artificial neural networks. Particle swarm optimization (PSO) is one of the most used tools for filtering out junk email. To make matters worse for ML techniques, datasets with millions of records tend to be quite challenging. In addition, dimensionality reduction and data preprocessing are essential for successful outcomes. These latter two steps need more focus and effort on the part of the processor. The main problem with ML methods is that, unlike deep learning methods, they can't pick up on details at a more basic level. One way in which neural networks (NN) are used in the field of machine learning (ML) is via deep learning (DL) [12]. Learning from abstract qualities does not need any data processing or dimensionality reduction, in contrast to ML approaches. The information is sent via a succession of hidden levels, where new information is added to that gathered in previous levels. In comparison to conventional ML algorithms, DL techniques have shown more reliable when dealing with massive datasets [13]. There are several applications for DL, including speech recognition, picture identification, and even drug discovery [14]. Two common DL approaches are recursive neural networks (RNNs) and convolutional neural networks (CNNs). Despite their recent use in detecting spam on social media [13], [15], [16], these techniques have not yet been widely adopted for use in identifying spam in electronic mail. In this study, we use the benefits of the RNN approach to fine-tune our spam email detector.

Related Works

Recent research has demonstrated that when compared to traditional machine learning techniques, deep learning is superior at identifying spam emails.

Mi et al. [17] used stacked auto-encoder (SAE) to harness the potential of DL for spam detection in email. Feature vectors are extracted from a set of test emails using methods like information gain (IG) and bag-of-words (BoW). The authors acknowledged that SAE's slow speed is one of its drawbacks. They suggested trying out several DL methods and digging into the parameter settings to significantly improve performance and cut down on runtime.

A spam detection algorithm based on deep learning was suggested by Chetty et al. [19]. The numerical and textual data are handled by separate architects in this deep model. A 57-node input layer, two 16-node hidden layers, a dropout layer, and a single-node output layer make up the initial architecture. The second design uses an embedding layer for words, followed by a pooling layer, a dense layer, and an output layer. In both designs, the dense layer activation function was a rectified linear unit (ReLu), whereas the Sigmoid function was employed for the output layer. The cross-entropy loss function was used in conjunction with the adaptive moment estimation (ADAM) optimizer. On the SpamBase dataset, the model (with the initial architecture) performed at a level of accuracy of 92.8% and an F1 score of 84.90%. The writers did not check the length of the film. While they acknowledged the current model's potential, they suggested more investigation into the parameter setting in order to identify the optimal values that would ultimately boost its performance. They also suggested trying out other DL frameworks.

Using a combination of a Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) and a Support Vector Machine (SVM), Alauthman [20] presented a method for detecting Bot spam emails. To begin, we used the CART feature reduction technique. The gradient issue in conventional RNN is then resolved using GRU, resulting in a shorter training time. The cross-entropy cost function and the ADAM optimizer were used by the writer. The SVM decision function is used to calculate the model's prediction at the final stage of the neural network. Using the SpamBase dataset, the method in this research achieved a detection rate of 98.7 per cent with the use of the smallest amount of attributes. However, the execution time was not presented, and the DL-related parameters were chosen at random without any experimental investigation. According to the author, the suggested model might be enhanced by including more ML methods.

Sumathi and Pugalendhi [21] presented a spam detection method that combines the strengths of a random forest with a deep neural network. The gini measure is used to determine which characteristics should be chosen by the random forest algorithm. The DNN classifier is subsequently trained on these characteristics. One hidden layer and 10 hidden nodes were employed in the backprobagation method. As a loss function, they used softmax. However, they didn't look into the DL parameters to see whether there were optimal settings. Based on the results of experiments conducted on the SpamBase dataset, DNN's classification rate beat K-NN and SVM with an accuracy of 88.59% when just the top five characteristics were taken into account. The writers did not check how long the film took to make. To improve accuracy, they suggested not experimenting with DL designs and settings but rather using bio-inspired feature selection techniques.

Hossain et al. [22] presented a model for detecting spam using ML and DL methods. Isolation Forest is initially applied to the dataset in order to eradicate any outliers. Random forest, KNN, and multinomial NB were the ML methods used. However, DL methods relied on single-layer RNNs for input and output. The authors did not specify a maximum number of hidden levels. When applied to the SpamBase dataset, RNN achieves an accuracy of 99.28%. The authors demonstrated that ML execution times might be drastically cut by narrowing the feature collection. While RNN showed promise, it was unable to prove this claim. As a means of spam detection, they advocated the use of ensemble learning.

Long Short-Term Memory (LSTM)

The connections between nodes in these networks, a special case of NN, have been modeled as a directed graph. The time series it's producing makes it possible to show how time evolves. RNNs can analyze input sequences of varying lengths by drawing on their memory, a feature made feasible by their foundation in feedforward neural networks. It prepares the way for them to take part in activities like unsegmented and sinter-connected thought. Typically-supported networks, which may be categorized into two subsets, have been accounted for using RNN. Situation in which one's natural urges are contained. On the other hand, another person feels a need that may sustain itself indefinitely. It has been shown that some kinds of networks display fleeting behaviour while being implemented. A recurrent neural network, or RNN, is a specific kind of artificial neural network (ANN) in computer science. As a result, the data fed into certain nodes may have been influenced by the data sent out of others. As a result, it is possible to display behaviors that change over time. A RNN's ability to maintain state means it can process input sequences of variable lengths. RNNs can understand sequences (use their memories) because of this capability. This makes them useful for tasks like voice or handwriting recognition that don't need segmentation. For all intents and purposes, recurrent neural networks are "Turing complete." This implies that they can process any input sequence using any technique. While a "CNN" can only process a limited amount of data at once, a "recurrent neural network" may potentially handle an infinite quantity of information. Both static and dynamic networks are not necessarily devoid of temporal dynamics. While a finite impulse recurrent network (IRN) may be unrolled and replaced by a strictly feed forward neural network, an infinite IRN, which is a directed cyclic graph, cannot. Both finite and infinite impulse recurrent networks may profit from the inclusion of stored states, and the neural network may manage the storage capacity independently. If the storage has delays or loops, another network or graph might be used instead. Examples of neural circuits that make use of such state control methods include long short-term memories (LSTMs) and gated recurrent units. In both cases, the term "remembered" refers to a gated state. A FNN is another name for it. The recurrent variant of ANN builds a graph from its nodes and the links between them. Because of this, time-domain demonstrations of dynamic behavior are now feasible. When compared to their feed forward ancestors, RNNs have no limits on the length of the data sequences they may analyze. They can recognize handwriting, but they may also be used to identify voices. Recurrent neural networks are Turing complete in the sense that they can analyze any sequences of data and run any programs.

Training LSTM Model with Data Classification and Prediction

Features for DL are produced from the original data by a hierarchical application of non-linear transformation operations. Because of this, it becomes clear what kind of DL network will be implemented. CNNs, DNNs, RNNs, and LSTM networks are only some of the most well-known types of deep learning. The literature provides sufficient data to conclude that DL algorithm performance is better than that of shallow learning methods. In comparison to traditional machine learning techniques, the computational cost of DL algorithms is substantially greater due to the massive quantity of data required during the training phase. Optimization of DL algorithms is a more nuanced process than optimization of traditional, "shallow," learning methods. The notion of transfer learning, which will be discussed in a moment, may help us get around these restrictions. A convolutional neural network (CNN) that has been trained using deep learning is running a classification task. In the realm of ML, DL is seen as a more advanced technique that may be used to extract features and give computers the ability to learn. To accomplish their goals, deep learning algorithms use a series of interconnected layers. The results from the most advanced layer have been routed to the lower layers so that they may be used as inputs. This method is greatly facilitating the extraction of characteristics in a hierarchical structure. Researchers in the field of deep learning are considering a wide variety of potential applications, including the processing of images and spoken language. In addition to its use in the medical area, CRM, and automobile automation also make use of the technology.

When the connections between the nodes of a network take the form of a directed graph, we say that the network is a recurrent neural network or a long short-term memory network. Additionally, it is producing a temporal sequence that allows it to display temporal dynamics. Some researchers in the area of DL consider LSTM [19] to be an ARNN architecture. The LSTM uses feedback connections, as shown in [20]. Several features set this feed forward neural network apart from the norm. It's not limited to single data types like photos, but can also process audio and video in the form of longer sequences.

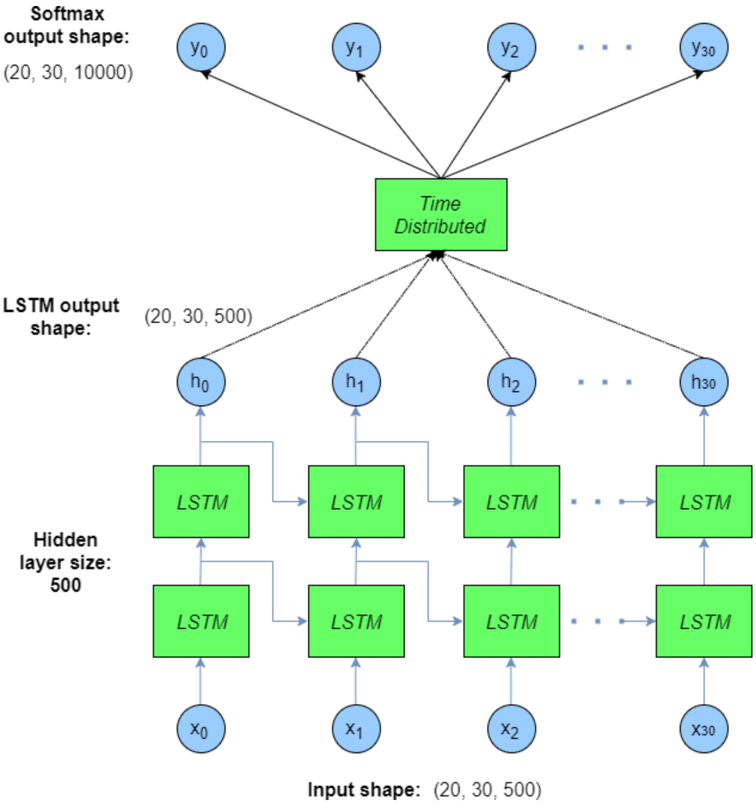

Figure 1. Working of LSTM

Text-to-speech synthesis and high-vocabulary voice recognition both benefited from the use of LSTM. In 2015, Google's voice recognition accuracy improved by 49% thanks to CTC-trained LSTM. Significant progress was made using LSTM in the areas of Language Modelling and Multilingual Language Processing. The accuracy of the CNNs used to generate picture captions was improved by including LSTM

Spam Detection Approaches

There are a number of techniques used to assess whether or not an incoming message is spam, including whitelist/blacklist, Bayesian analysis, Mail header analysis, Keyword testing, etc. Some of their justifications are listed below:

- Whitelist/Blacklist: - These methods do little more than generate a list. A user's whitelist consists of safe contacts, such as known email addresses or websites. Users may also make use of an automated white list administration application to expeditiously add trusted IP addresses to the approved list. The opposite of a "whitelist" is a "blacklist." We add potentially dangerous Internet Protocol addresses here.

• Mail Header Checking: - This method has a lot of name recognition. In this, we only use a set of criteria to compare against email headers. Emails with invalid headers (such as an empty "From" or "To" field or an address with an excessive number of digits) are sent back to the sender.

• Signatures: - A signature with a different hash value is generated for each spam message using this method. The filters check incoming emails against a database of historical values. It's quite unlikely that a genuine communication would have the same value as a piece of spam that was previously kept.

• Bayesian Classifier: - Words used in spam emails differ from those used in legitimate emails. There is a heightened likelihood that these terms will appear in both correspondences. These probabilities are not hardwired into the filters we used; rather, they need training before they can be utilized. After training, the word probabilities are used to determine the likelihood that a given email with a certain collection of terms is spam or not. Whether an email is more likely to be spam depending on whether it contains every possible word or only the most intriguing terms. Bayes' theorem is used to calculate this contribution, which is called the posterior probability. The emails' spam likelihood is then calculated for each and every word. Emails will be flagged as spam if their overall value is more than a certain limit.

SpamBase is a dataset available in the UCI Machine Learning Repository that was used to train the RNN in this work. The dataset includes 4500 emails with 60 different characteristics. The data was trained and tested using the open-source program deep learning studio, which is designed for developing deep learning networks. As can be seen in Figure 2, several experiments are conducted by varying the activation function, the dropout rate, and the number of epochs.

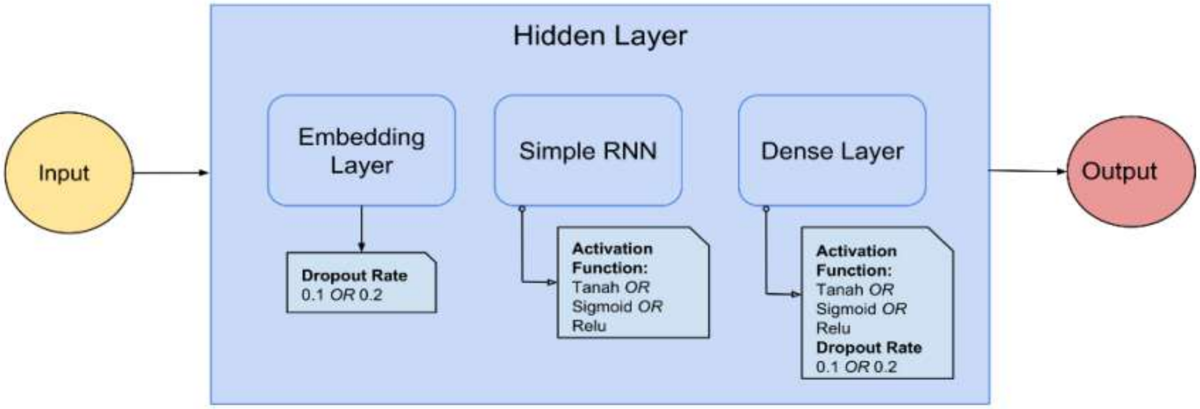

Figure 2. RNN Structure used in this study

The ideal parameters for the experiment have been determined. There are an equal number of neurons representing features in the input layer. There are three rounds total, and each round consists of four runs. The activation function is kept constant throughout iterations, but the number of epochs and the dropout % are changed. Initially, we clamp down on the Tanh activation function. Then, we use a dropout of 0.1 and a range of 100–150 epochs for the first two iterations. Alter the dropout rate for the remaining runs in the first round to 0.5, and boost the epoch count for each runs, respectively, to 100 and 150. In Rounds 3 and 4, the same procedures are repeated, but a different activation function is used. A total of 80% of the data is reserved for use in the training set, while 10% each are used for validation and testing. We keep track of the accuracy rates in both training and validation for each cycle. Exam precision is only guaranteed in the best-case situation. The proposed model's greatest accuracy is measured against that of other research that have used the same dataset.

Conclusion

In this research, we use the accuracy of deep learning methods to identify junk email. Many different activation functions, epoch counts, and dropout rates are used in RNN applications. Tanh as the activation function, 0.1 as the dropout rate, and 100 as the number of epochs are the ideal settings for this experiment. Other research that have used the SpamBase database are compared to the suggested technique as well. However, the suggested RNN has shown to be more effective than the existing state-of-the-art method, which combines a Gated Recurrent Unit Recurrent Neural Network with SVM to attain an accuracy of 98.7 percent using as little features as possible. Using long-short term memory (LSTM), the team plans to enhance spam email categorization in future research. LSTM is one kind of RNN that has shown to be quite effective.

References

[1] J. Johnson, “Number of sent and received e-mails per day worldwide from 2017 to 2025.” April, 2021, Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/

[2] T. Kulikova, T. Shcherbakova, and T. Sidorina, “Spam and phishing in Q1 2021 | Securelist.” 2021, Accessed: Jan. 20, 2022. [Online]. Available: https://securelist.com/spam-and-phishing-in-q1-2021/102018/

[3] N. M. Samsudin, C. F. B. Mohd Foozy, N. Alias, P. Shamala, N. F. Othman, and W. I. S. Wan Din, “Youtube spam detection framework using naïve bayes and logistic regression,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 14, no. 3, pp. 1508–1517, Jun. 2019, doi: 10.11591/ijeecs.v14.i3.pp1508-1517.

[4] N. Alias, C. F. M. Foozy, and S. N. Ramli, “Video spam comment features selection using machine learning techniques,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 15, no. 2, pp. 1046–1053, Aug. 2019, doi: 10.11591/ijeecs.v15.i2.pp1046-1053.

[5] A. Al-Ajeli, R. Alubady, and E. S. Al-Shamery, “Improving spam email detection using hybrid feature selection and sequential minimal optimisation,” Indonesian Journal of Electrical

Engineering and Computer Science, vol. 19, no. 1, pp. 535–542, Jul. 2020, doi: 10.11591/ijeecs.v19.i1.pp535-542.

[6] A. Sharaff, N. K. Nagwani, and A. Dhadse, "Comparative Study of Classification Algorithms for Spam Email Detection," in Emerging Research in Computing, Information, Communication and Applications, Springer India, pp. 237–244, 2016.

[7] S. M. Abdulhamid, M. Shuaib, O. Osho, I. Ismaila, and J. K. Alhassan, "Comparative Analysis of Classification Algorithms for Email Spam Detection," International Journal of Computer Network and Information Security, vol. 10, no. 1, pp. 60–67, Jan. 2018, doi: 10.5815/ijcnis.2018.01.07.

[8] H. Kaur and A. Sharma, "Novel Email Spam Classification using Integrated Particle Swarm Optimization and J48," International Journal of Computer Applications, vol. 149, no. 7, pp. 23–27, Sep. 2016, doi: 10.5120/ijca2016911466.

[9] A. Rodan, H. Faris, and J. Alqatawna, "Optimizing Feedforward Neural Networks Using Biogeography Based Optimization for E-Mail Spam Identification," International Journal of Communications, Network and System Sciences, vol. 09, no. 01, pp. 19–28, 2016, doi: 10.4236/ijcns.2016.91002.

[10] 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, pp. 68–74, Jul. 2016, doi: 10.5815/ijmecs.2016.07.08.

[11] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," SIAM Review, vol. 60, no. 2, pp. 223–311, Jan. 2018, doi: 10.1137/16M1080173.

[12] J. Schmidhuber, "Deep Learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[13] G. Jain and B. Agarwal, "An Overview of RNN and CNN Techniques for Spam Detection in Social Media," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 10, p. 2277, 2016.

[14] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[15] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic LSTM for spam detection," International Journal of Information Technology (Singapore), vol. 11, no. 2, pp. 239–250, Apr. 2019, doi: 10.1007/s41870-018-0157-5.

[16] G. Jain, M. Sharma, and B. Agarwal, “Spam detection in social media using convolutional and long short term memory neural network,” Annals of Mathematics and Artificial Intelligence, vol. 85, no. 1, pp. 21–44, Jan. 2019, doi: 10.1007/s10472-018-9612-z.

[17] G. Mi, Y. Gao, and Y. Tan, “Apply stacked auto-encoder to spam detection,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9141, Springer International Publishing, pp. 3–15, 2015.

[18] A. Barushka and P. Hajek, “Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks,” Applied Intelligence, vol. 48, no. 10, pp. 3538–3556, Mar. 2018, doi: 10.1007/s10489-018-1161-y.

[19] G. Chetty, H. Bui, and M. White, “Deep learning based spam detection system,” in Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2019, Dec. 2019, pp. 91–96, doi: 10.1109/iCMLDE49015.2019.00027.

[20] M. Alauthman, “Botnet spam e-mail detection using deep recurrent neural network,” International Journal of Emerging Trends in Engineering Research, vol. 8, no. 5, pp. 1979–1986, May 2020, doi: 10.30534/ijeter/2020/83852020.

[21] S. Sumathi and G. K. Pugalendhi, “Cognition based spam mail text analysis using combined approach of deep neural network classifier and random forest,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 6, pp. 5721–5731, May 2021, doi: 10.1007/s12652-020-02087-8.

[22] F. Hossain, M. N. Uddin, and R. K. Halder, “Analysis of optimized machine learning and deep learning techniques for spam detection,” Apr. 2021, doi: 10.1109/IEMTRONICS52119.2021.9422508.

[23] I. AbdulNabi and Q. Yaseen, “Spam email detection using deep learning techniques,” Procedia Computer Science, vol. 184, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.