Advancing Email Spam Classification using Machine Learning and Deep Learning Techniques

Meaad Hamad Alsuwit
Department of Information Technology, College of Computer Sciences and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
431204675@s.mu.edu.sa

Mohd Anul Haq
Department of Computer Science, College of Computer Sciences and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
m.anul@mu.edu.sa

Mohammed A. Aleisa
Department of Computer Science, College of Computer Sciences and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
m.aleisa@mu.edu.sa (corresponding author)

Received: 25 April 2024 | Revised: 13 May 2024 | Accepted: 17 May 2024

ABSTRACT

Email communication has become integral to various industries, but the pervasive issue of spam emails poses significant challenges for service providers. This research proposes a study leveraging Machine Learning (ML) and Deep Learning (DL) techniques to effectively classify spam emails. Methods such as Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF), and Artificial Neural Networks (ANNs) are employed to construct robust models for accurate spam detection. By amalgamating these techniques, the aim is to enhance efficiency and precision in spam detection, aiding email and IoT service providers in mitigating the detrimental effects of spam. Evaluation of the proposed models revealed promising outcomes. LR, RF, and NB achieved an impressive accuracy of 97% and an F1-Score of 97.5%, showcasing their efficacy in accurately identifying spam emails. The ANN model demonstrated slightly superior performance, with 98% accuracy and 97.5% F1-score, suggesting potential improvements in accuracy and robustness in spam filtering systems. These findings underscore the viability of both traditional ML algorithms and DL approaches in addressing the challenges of email spam classification, paving the way for more effective spam detection mechanisms in electronic communication platforms.

Keywords-spam; ML; DL; spam classification; email

I. INTRODUCTION

As internet communication expanded, the email has emerged as a trusted and effective tool. Nevertheless, it has also attracted the attention of marketing companies and individuals posing online threats. Spam, or unsolicited email, is sent in bulk to numerous recipients without their agreement [1]. Typically dispatched by marketers to advertise products or services, these emails can also come from parties with harmful motives, like orchestrating phishing schemes or disseminating harmful software. The widespread issue of spam and phishing has created significant challenges for both individuals and businesses, causing financial damage and breaches of privacy, often due to insufficient knowledge about online safety and inadequate email filtering systems. In 2022, a staggering 55% of all emails were classified as spam, translating to roughly 15.4 billion unwanted messages bombarding inboxes daily. This spam epidemic costs internet users an estimated $355 million annually [2]. While email service providers offer some defense with spam filters, vigilance is crucial. We must carefully scrutinize emails before opening or clicking embedded links. The battle against spam is ongoing. Spammers constantly adapt their tactics to bypass detection. One technique involves using seemingly legitimate email addresses, making them appear trustworthy at first glance [3]. Additionally, spam that incorporates personalized details, like your name, profession, or other private information, can bypass filters designed to identify generic spam messages [4]. This personalization makes it even more critical to be cautious and verify email legitimacy before taking any action [5].

The task of classifying spam emails is a dynamic and complex issue, with numerous ML strategies being extensively researched to enhance their precision and effectiveness. Many previous research efforts have delved into various facets of spam email classification, covering topics like the deployment of ML methods, adversarial strategies, the incorporation of ensemble techniques, and the application of unsupervised learning. In [6], a comparative analysis of various ML algorithms for spam classification was performed. The authors investigated classification models such as the Support Vector Machine (SVM) classifier, k-Nearest Neighbor (kNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), AdaBoost classifier, and bagging classifier. Their results indicated that the SVM and kNN classifiers achieved a precision of 0.92, NB obtained 0.87, the DT scored 0.94, RF achieved 0.90, the AdaBoost classifier reached 0.95, and the Bagging classifier obtained 0.94. Authors in [7] conducted a series of experiments utilizing the Enron dataset, where they evaluated the performance of five distinct classification algorithms [7]. The employed algorithms were SVM, RF, NB, DT, and kNN, achieving accuracies of 97.83%, 97.60%, 95.48%, 90.90%, and 95.29% respectively. Their findings highlighted SVM as the most effective classifier, closely followed by the RF classifier. Additionally, the researchers proposed future research avenues aimed at enhancing accuracy through the utilization of more computationally intensive, yet highly accurate ensemble methods such as XGboost. In [8], the authors evaluated the effectiveness of six ML techniques for spam classification, utilizing the SpamAssassin dataset. The NB method demonstrated remarkable accuracy, achieving 99.46%, while SVM attained 96.90% accuracy, and the kNN algorithm exhibited 96.20% accuracy. Furthermore, the ANN approach yielded an accuracy of 96.83%, the artificial immune system recorded 96.23%, and the rough sets method showed an accuracy of 97.42%. In [9], the focus was on adversarial methods employed to evade spam email classification techniques, along with the exploration of countermeasures against such attacks. The authors also discussed the limitations of current methods and proposed guidelines for future research in spam email classification.

Authors in [10] conducted a study investigating multiple ML methods for spam email classification, incorporating DT, SVM, and NB classifiers [10]. Their findings suggested that SVM and DT exhibited comparable performance, particularly in handling emails with extensive content, such as those surpassing 10,000 words. [11, 12]. Authors in [13] developed a DL model that utilized features like character n-grams and word embeddings, alongside an unsupervised topic modeling technique, for addressing a similar challenge. Their investigation yielded promising results, outperforming several existing baseline ML models. Similarly, authors in [14] explored an unsupervised topic modeling technique for spam email classification, achieving comparable outcomes. They utilized the latent Dirichlet allocation model to generate features from the training data, which were then utilized in a DL model. In a unique approach, authors in [15] combined CNN and LSTM methods for email classification, surpassing traditional techniques such as Gaussian NB and DT [15]. Authors in [16] conducted a comprehensive study, assessing the strengths and weaknesses of various ML models in spam email classification, including an exploration of hyperparameter tuning [16]. Authors in [17] merged DT and RF classifiers to improve classification accuracy, demonstrating enhanced results [17]. Additionally, authors in [18] enhanced the precision of identifying spam in online reviews by employing a stacking approach, achieving notable outcome. Authors in [19] introduced the Fast Adaptive Stacking of Ensembles (FASE) method tailored for learning from non-stationary data streams. This algorithm was capable of processing real-time data with consistent time and space complexity, resulting in a significant enhancement in predictive accuracy compared to several conventional ML techniques. Moreover, authors in [20] implemented a stacking method integrating NB, SVM, DT, and a meta-classifier for email spam classification, achieving a precision rate of 95.67%. Authors in [21] utilized a stacking-based CNN to detect fraudulent or spam tweets.

In [22], the authors developed a strategy to detect spam comments on YouTube, emphasizing the need for more efficient spam detection methods beyond YouTube's existing systems. They conducted tests with six different ML methods and two ensemble models on comment data from popular videos, contributing to the improvement of spam detection on YouTube and addressing related challenges. Authors in [23] focused on spam identification in social media networks, proposing a heterogeneous stacking-based ensemble learning framework to counter the issue of class imbalance. Their experimental results demonstrated enhanced spam detection in imbalanced datasets, thereby improving information security in social networks. Authors in [24] addressed the problem of class imbalance in Twitter spam detection by introducing a fuzzy-based oversampling method named FOS. They developed an ensemble learning strategy that involved adjusting the class distribution, creating classification models on redistributed datasets, and aggregating predictions through majority voting. Their experiments showed a significant boost in the spam detection rate for imbalanced class distributions, effectively reducing Twitter spam. In [25], focus was given on spam email detection and classification in the realm of cybersecurity. Standard models utilizing RF and XGBoost ensemble algorithms were developed, along with hyperparameter optimization techniques. The refined XGBoost model outperformed the RF model, demonstrating superior accuracy, sensitivity, and F1 scores. This enhanced XGBoost model proved effective and efficient in recognizing spam emails, making significant contributions to cybersecurity measures. Additionally, the researchers emphasized the importance of maintaining the reliability of software and code for quality research in classification problems [26–28].

The studies discussed above showcase the diverse range of methods and advancements made in spam email classification using ML techniques. The current research offers a crucial solution to the challenge of spam emails by leveraging a blend of Machine Learning (ML) and Deep Learning (DL) techniques. By combining traditional algorithms like Logistic Regression (LR) and NB with advanced ANNs, robust models for accurate spam detection are constructed. Through meticulous tuning and evaluation, the proposed approach achieves impressive values of precision, recall, and F1 score, highlighting its efficacy in identifying spam emails. This comprehensive method not only enhances efficiency, but also provides valuable insights for improving spam filtering mechanisms, offering a promising avenue for email and IoT service providers to combat the detrimental effects of spam.

II. METHOD

A. Dataset Description and Preprocessing

This study utilized two authentic datasets for classifying spam emails. The primary dataset, presented as a comprehensive CSV file, consisted of 83,446 email records categorized as spam (labeled with "1") or nonspam ("0"). This dataset was created by merging the 2007 TREC Public Spam Corpus and the Enron-Spam Dataset [29, 30]. The dataset consisted of 83,448 entries organized into two columns: 'label' and 'text'. The 'label' column contained integer values, and all entries were non-null. The 'text' column contained email text data and was also free of null values. The data types for these columns were 'int64' and 'object', respectively, and the dataset consumed approximately 140 MB of memory. The label distribution showed a relatively balanced dataset, with Class 1 having a slightly higher count than Class 0. Consequently, oversampling techniques like SMOTE may have been less necessary due to the moderate class imbalance [1].

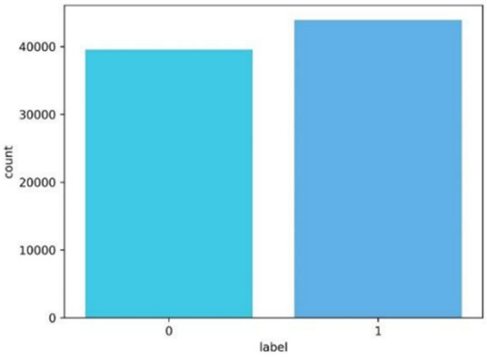

Fig. 1. The label information for the dataset used.

This study utilized the sklearn library to construct ML models, encompassing LR, NB, RF, and tf.keras library to develop the ANN model. Various tools from sklearn were employed for tasks such as data preparation, model training, and evaluation. Additionally, essential libraries like pandas and NumPy were seamlessly integrated to effectively manage data manipulation and numerical computations. For the creation of informative visualizations, Matplotlib was employed, while Seaborn played a role in enhancing the visual appeal of statistical graphics.

B. Logistic Regression

In the initial phase of the analysis, LR, a widely-used method for predictive analytics and classification tasks, was applied. To transform the odds, which is the probability of success divided by the probability of failure, the logit formula was employed:

$$Logit(p) = \frac{1}{1+\exp(-p)} \tag{1}$$

The function of *Logit*(*p*) in LR is to transform the odds of success to a linear scale, facilitating binary classification by modeling the probability of the outcome:

$$\ln\frac{p}{1-p} = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k \tag{2}$$

where p is the probability of an event, X_1, …,X_k are predictor variables, and $\beta_0, \beta_1, . . . , \beta_k$ are coefficients that determine the impact of each predictor variable.

C. Naive Bayes

The NB classifiers denote a collection of methods grounded in Bayes' theorem. NB represents a range of algorithms, which operate on a shared fundamental principle. Each set of characteristics being sorted into categories is unique from others. Generally, the assumptions made by NB classifiers are not entirely accurate in real-world situations. However, in practical terms, although the modeling technique is not perfect, it often performs well. Bayes' theorem calculates the probability of one event happening based on the probability of another event occurring.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{3}$$

D. Random Forest

In RF, each DT in the group is made using a sample of data picked from a larger set. This method, which extends the bagging technique, creates a bunch of DTs that work together. Unlike a single DT, RF uses feature randomness, meaning it picks only some features to consider, preventing them from being too similar. The final prediction for an observation in RF is then decided by seeing which class got the most votes from all the trees. So, for one tree:

$$\hat{y}_i = (yi1, yi2, \ldots, yim) \tag{4}$$

where, $\hat{yi}$ is the predicted class for the *i*-th observation, and yij is the predicted class by the *j*-th DT. For the whole RF:

$$\hat{Y} = (\hat{y}_1, \hat{y}_2, \ldots\ldots, \hat{y}_k) \tag{5}$$

This way, by combining predictions from many DTs, RF makes more accurate and reliable predictions compared to just one DT.

III. IMPLEMENTATION

A. DL Model Development

In this research, an embedding method was applied using the TensorFlow Hub library. The hub_layer variable was instantiated as a Keras layer through the hub. The Keras layer was configured to handle string data types and was designated as trainable during the model training process. This embedding layer played a pivotal role in capturing the semantic representations of textual data within the ANN architecture. In this research, a sequential neural network model was designed using TensorFlow's Keras framework. The model architecture consisted of an initial embedding layer employing a new ANN model to capture meaningful representations of textual data. Two subsequent dense layers with 32 units each and Rectified Linear Unit (ReLU) activation function were introduced, accompanied by batch normalization for improved training stability. Dropout regularization with a rate of 0.4 was strategically incorporated after each dense layer to mitigate overfitting concerns. The final layer featured a single-unit dense layer with a sigmoid activation function, tailored for binary classification tasks. To optimize the training process, a learning rate schedule was implemented using exponential decay, with an initial learning rate of 0.001, decay steps set at 10,000, and a decay rate of 0.9. The Adam optimizer, configured with the specified learning rate schedule, was employed. The model was then compiled using the binary cross entropy loss function, suitable for binary classification, and accuracy as the chosen evaluation metric. This comprehensive configuration aimed to facilitate effective learning from textual data while employing regularization techniques and learning rate scheduling to enhance model generalization and prevent overfitting during the training phase.

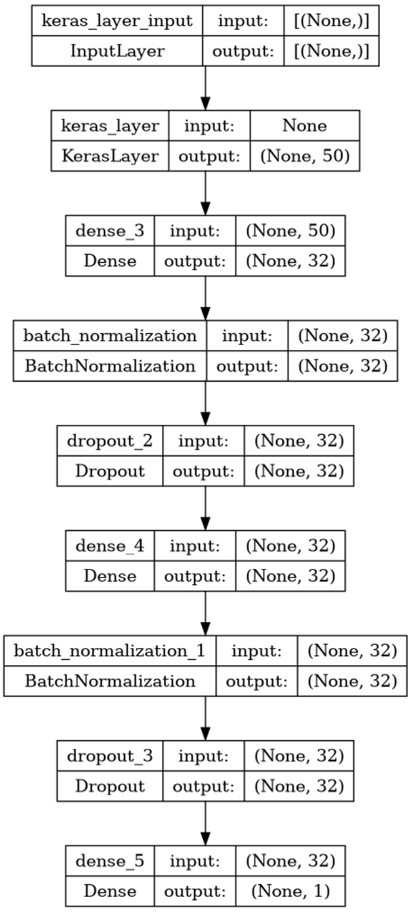
keras_layer_input | input: | [(None,)]
InputLayer | output: | [(None,)]
keras_layer | input: | None
KerasLayer | output: | (None, 50)
dense_3 | input: | (None, 50)
Dense | output: | (None, 32)
batch_normalization | input: | (None, 32)
BatchNormalization | output: | (None, 32)
dropout_2 | input: | (None, 32)
Dropout | output: | (None, 32)
dense_4 | input: | (None, 32)
Dense | output: | (None, 32)
batch_normalization_1 | input: | (None, 32)
BatchNormalization | output: | (None, 32)
dropout_3 | input: | (None, 32)
Dropout | output: | (None, 32)
dense_5 | input: | (None, 32)
Dense | output: | (None, 1)

Fig. 2. The developed ANN architecture for spam detection.

This computation made use of a single GPU (i9, 10900k, 128 GB 2666MHz RAM) for enhanced processing capabilities. The architecture of the ANN model, which has been recommended for employment in malware detection, is visually represented in Figure 2.

B. Hyperparameter Tuning

Hyperparameter tuning is essential to optimize model performance by fine-tuning parameters, ensuring the best configuration for accurate predictions and improved overall effectiveness. The hyperparameters for each model were meticulously adjusted to improve performance. For the LR model, the regularization penalty was set to 'elasticnet' and the regularization strength (C) was chosen as 0.01. The solver method 'lbfgs' was utilized with a maximum of 500 iterations. In the RF model, key hyperparameters such as the number of estimators (n_estimators), maximum depth of trees, minimum samples required for a split (min_samples_split), and minimum samples required at each leaf node (min_samples_leaf) were fine-tuned to 150, 10, 5, and 2 respectively, and the maximum number of features (max_features) was adjusted to 'auto'. For the NB model, the Laplace smoothing parameter (alpha) was carefully set to 0.1. Finally, an ANN model was constructed with 32 neurons, a dropout rate of 0.4, an initial learning rate of 0.001, a decay step of 10,000, and a decay rate of 0.9. This detailed process of hyperparameter tuning aimed to optimize model performance and achieve superior predictive accuracy. The present study used GridSearchCV, a greedy optimization-based approach, to tune hyperparameters due to its systematic and intensive search abilities, ensuring the best model performance by searching a wide range of parameter combinations.

C. Evaluation Measures

In this section, the performance of ML models is evaluated using four metrics: precision, recall, F1-score, and accuracy. Accuracy reflects the proportion of emails correctly identified as spam out of the total number of regular emails. Recall is defined as the ratio of identified spam messages to the total number of spam samples, while precision is the ratio of correctly identified spam messages to the total number of email messages identified as spam.

$$\text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{6}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}} \tag{7}$$

$$\text{Accuracy}=\frac{\text{TP}}{\text{TP+FN}} \tag{8}$$

$$\text{F1-score}=\frac{\text{Precision+Recall}}{\text{Precision}\times\text{Recall}} \tag{9}$$

The number of correctly classified spam is referred to as the True Positive (TP). True Negative (TN) represents the number of legitimate emails that have been correctly classified as not spam. The number of spam messages misclassified as legitimate emails is referred to as the False Negative (FN). The number of legitimate emails misclassified as spam is referred to as False Positives (FP).

TABLE I. THE TUNED HYPERPARAMETERS OF ALL MODELS

Model	Hyperparameters	Tuned hyperparameters
LR	penalty: [l1, l2, elasticnet] C: [0.001, 0.01, 0.1, 1, 10, 100] solver: [newton-cg, lbfgs, saga, liblinear] max_iter: [100, 500, 1000]	Elasticeet 0.01 Lbfgs 500
RF	n_estimators: [50, 100, 150, 200] max_depth: [None, 10, 20] min_samples_split: [2, 5, 10] min_samples_leaf: [1, 2, 4] max_features: [auto, sqrt, log2]	150 10 5 2 Auto
NB	alpha:[0.01, 0.1, 0.2, 1]	0.1
ANN	neurons: [16, 32, 64, 128, 256] dropout_rate: [0.2, 0.4, 0.6, 0.8] initial_learning_rate: [0.001, 0.01, 0.1] decay_steps: [5000, , 15000] decay_rate: [0.8, 0.9, 0.95]	32 0.4 0.001 10000 0.9

IV. RESULTS AND DISCUSSION

The evaluation of ML and DL techniques for email spam classification yielded promising outcomes, which can be seen in Table II. LR, RF, and NB achieved an impressive average precision, recall, and F1-score of 97.5%. These traditional methods demonstrated strong performance, indicating their efficacy in accurately identifying spam emails. Additionally, the ANN model showcased slightly superior performance, with an average precision, recall, and F1-score also at 97.5%. This suggests that DL techniques offer potential improvements in email spam classification, potentially enhancing accuracy and robustness in spam filtering systems. These results underscore the viability of both traditional ML algorithms and DL approaches in addressing the challenges of email spam classification, paving the way for more effective spam detection mechanisms in electronic communication platforms. The confusion matrices of the considered classifiers are shown in Figures 3-7.

TABLE II. MODEL PERFORMANCE

Model	Precision	Recall	F1-score	Accuracy
LR	97.50%	97.50%	97.50%	97%
RF	97.50%	97.50%	97.50%	97%
NB	96.50%	96.50%	96.50%	97%
NN	97.50%	97.50%	97.50%	98%

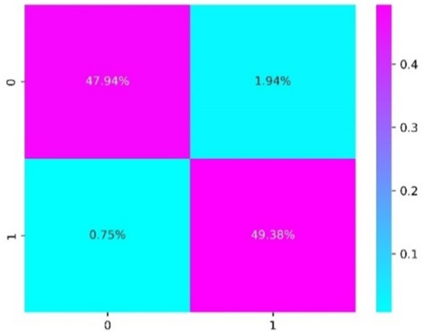

Fig. 3. Confusion matrix of the LR model.

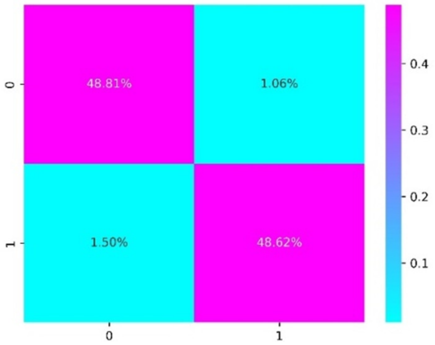

Fig. 4. Confusion matrix of the RF model.

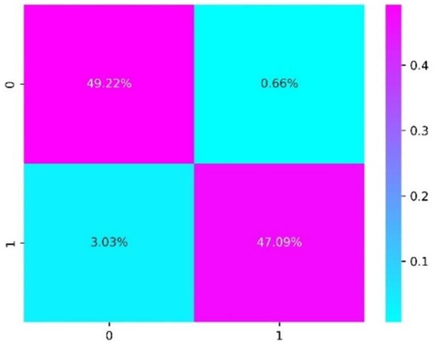

Fig. 5. Confusion matrix of the NB model.

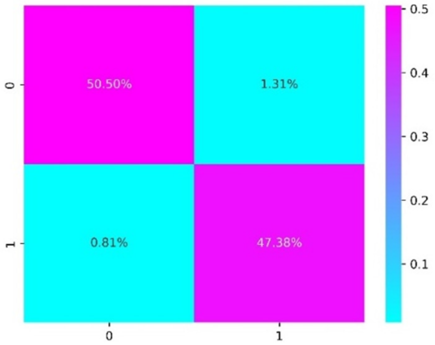

Fig. 6. Confusion matrix of the ANN model.

In the context of spam classification with the use of ANNs, Accuracy and Loss Curves are fundamental visualizations for gauging model performance during training. The Accuracy Curve showcases the model's ability to correctly classify spam and nonspam instances over epochs, offering insights into the learning process. On the other hand, the Loss Curve depicts the diminishing training loss over time, providing an understanding of how well the model is minimizing errors. These curves aid researchers and practitioners in determining the optimal number of epochs, identifying convergence, and ensuring the model's proficiency in discriminating between spam and non-spam emails. The ROC (Receiver Operating Characteristic) curve visually represents the trade-off between the TP rate (sensitivity) and the FP rate (specificity) at different threshold values. It illustrates how well a model distinguishes between true positives and false positives across various threshold settings. The area under the ROC curve (AUC-ROC) quantifies the model's ability to discriminate between positive and negative instances. A higher AUC-ROC value, closer to 1, indicates better discriminatory power, suggesting that the model performs well.

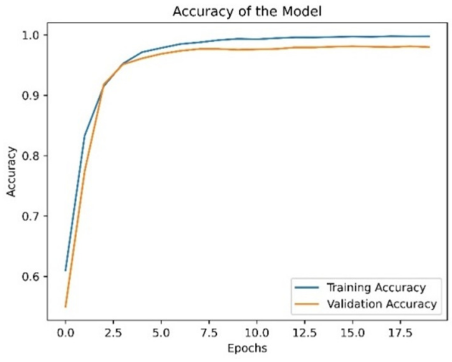

Fig. 7. Accuracy curve for the ANN model.

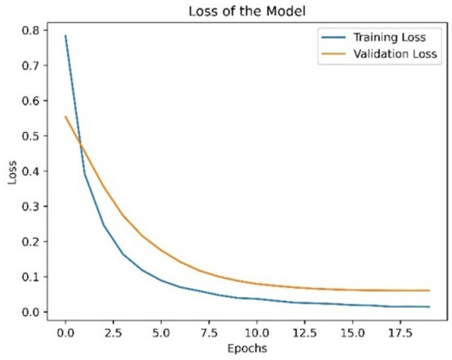

Fig. 8. Loss curve for the ANN model.

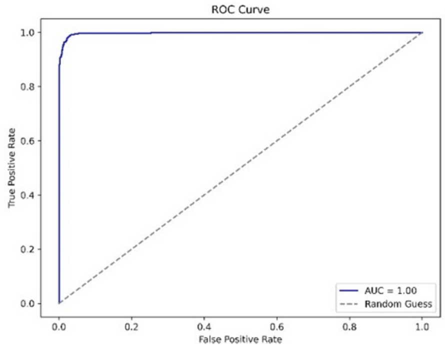

Fig. 9. ROC curve for the ANN model.

Table III shows a comparison of this studies results with other, relevant studies employing SVM, kNN, NB, DT, RF, AdaBoost, bagging [6], SVM, RF, NB, DT, kNN [7], SVM, kNN, ANN, AIS, RS [8], stacking NB, SVM, DT [20], stacking CNN [21], FOS and ensemble learning [26], and RF and XGBoost [27]. It can be seen that the acquired results from this study either surpass the others or are a little behind.

TABLE III. COMPARISON WITH OTHER STUDIES

Ref.	Year	Models used	Key Metrics
[6]	2020	SVM, kNN, NB, DT, RF, AdaBoost, Bagging	Precision: SVM (0.92), kNN (0.92), NB (0.87), DT (0.94), RF (0.90), AdaBoost (0.95), Bagging (0.94)
[7]	2021	SVM, RF, NB, DT, kNN	Accuracy: SVM (97.83%), RF (97.60%), NB (95.48%), DT (90.90%), kNN (95.29%)
[8]		SVM, KNN, ANN, AIS, RS	SVM (96.90%), kNN (96.20%), ANN (96.83%), AIS (96.23%), RS (97.42%)
[20]	2017	Stacking approach with NB, SVM, DT, meta-classifier	Precision: 95.67%
[21]	2021	Stacking based CNN	Detecting fake or spam tweets
[26]	2023	FOS and ensemble learning	Significant improvement in spam detection on Twitter
[27]	2023	RF, XGBoost	XGBoost model showing superior performance over RF
Current Study	2024	LR RF NB NN	Accuracy 97% 97% 97% 98%

V. CONCLUSIONS

The proposed method significantly improved the accuracy of spam email classification, by leveraging ML algorithms. Through the experiments conducted, it was observed that integrating the outputs of multiple base classifiers led to enhancements in precision, recall, and F1-score values. These findings suggest that the ML holds promise for effectively enhancing the accuracy of spam email classification in practical applications. Looking ahead, datasets incorporating both images and personalized email content could yield more promising results. Initially, gathering and organizing relevant data is crucial for addressing this challenge. Additionally, another area for future exploration is the classification of spam emails that have undergone processing through email warming tools. These tools aim to circumvent email servers or algorithms by sending simulated emails to establish a positive sending reputation with email providers, thereby decreasing the likelihood of subsequent emails from the sender being marked as spam. Emails influenced by such tools could potentially impact the accuracy of spam email classifiers. Therefore, obtaining a dataset containing these emails would be very useful [31-36]. Further research and development are required to validate these results and explore additional advantages of employing the advanced method in various classification scenarios [37-41].

ACKNOWLEDGEMENTS

The authors extend their appreciation to the Deanship of Postgraduate Studies and Scientific Research at Majmaah University for funding this research work through the project number (PGR-2024-1122).

REFERENCES

[1] S. L. Pfleeger and G. Bloom, "Canning SPAM: Proposed solutions to unwanted email," *IEEE Security & Privacy*, vol. 3, no. 2, pp. 40–47, Mar. 2005, https://doi.org/10.1109/MSP.2005.38.

[2] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *17th ACM Conference on Computer and Communications Security*, Chicago, IL, USA, Oct. 2010, pp. 27–37, https://doi.org/10.1145/1866307.1866311.

[3] D. Kumar and R. Kumar, "Spam Filtering using SVM with different Kernel Functions," *International Journal of Computer Applications*, vol. 136, no. 5, pp. 16–23, Feb. 2016, https://doi.org/10.5120/ijca2016908395.

[4] R. Heartfield and G. Loukas, "A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks," *ACM Computing Surveys*, vol. 48, no. 3, Sep. 2015, Art. no. 37, https://doi.org/10.1145/2835375.

[5] J. John, A. Moshchuk, S. Gribble, and A. Krishnamurthy, "Studying Spamming Botnets Using Botlab," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, Boston, MA, USA, Jan. 2009, pp. 291–306.

[6] N. Kumar, S. Sonowal, and Nishant, "Email Spam Detection Using Machine Learning Algorithms," in *Second International Conference on Inventive Research in Computing Applications*, Coimbatore, India, Jul. 2020, pp. 108–113, https://doi.org/10.1109/ICIRCA48905.2020.9183098.

[7] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, and D. Karia, "E-Mail Spam Classification via Machine Learning and Natural Language Processing," in *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks*, Tirunelveli, India, Feb. 2021, pp. 693–699, https://doi.org/10.1109/ICICV50876.2021.9388530.

[8] W. A. Awad and S. M. ELseuofi, "Machine Learning Methods for Spam E-Mail Classification," *International Journal of Computer Science and Information Technology*, vol. 3, no. 1, pp. 173–184, Feb. 2011, https://doi.org/10.5121/ijcsit.2011.3112.

[9] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial Feature Selection Against Evasion Attacks," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 766–777, Mar. 2016, https://doi.org/10.1109/TCYB.2015.2415032.

[10] K. Shaukat, S. Luo, S. Chen, and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," in *International Conference on Cyber Warfare and Security*, Islamabad, Pakistan, Oct. 2020, pp. 1–6, https://doi.org/10.1109/ICCWS48432.2020.9292388.

[11] A. Garavand, C. Salehnasab, A. Behmanesh, N. Aslani, A. H. Zadeh, and M. Ghaderzadeh, "Efficient Model for Coronary Artery Disease Diagnosis: A Comparative Study of Several Machine Learning Algorithms," *Journal of Healthcare Engineering*, vol. 2022, Oct. 2022, Art. no. 5359540, https://doi.org/10.1155/2022/5359540.

[12] M. Ghaderzadeh, M. Aria, and F. Asadi, "X-Ray Equipped with Artificial Intelligence: Changing the COVID-19 Diagnostic Paradigm during the Pandemic," *BioMed Research International*, vol. 2021, Aug. 2021, Art. no. e9942873, https://doi.org/10.1155/2021/9942873.

[13] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17259–17274, Dec. 2020, https://doi.org/10.1007/s00521-020-04757-2.

[14] V. Ramanathan and H. Wechsler, "Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation," *Computers & Security*, vol. 34, pp. 123–139, May 2013, https://doi.org/10.1016/j.cose.2012.12.002.

[15] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages," *Future Internet*, vol. 12, no. 9, Sep. 2020, Art. no. 156, https://doi.org/10.3390/fi12090156.

[16] M. V. Madhavan, S. Pande, P. Umekar, T. Mahore, and D. Kalyankar, "Comparative Analysis of Detection of Email Spam With the Aid of Machine Learning Approaches," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, Jan. 2021, Art. no. 012113, https://doi.org/10.1088/1757-899X/1022/1/012113.

[17] A. Rayan, "Analysis of e-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique," *Computational Intelligence and Neuroscience*, vol. 2022, Aug. 2022, Art. no. e2500772, https://doi.org/10.1155/2022/2500772.

[18] A. K. Suborna, S. Saha, C. Roy, S. Sarkar, and Md. T. H. Siddique, "An Approach to Improve the Accuracy of Detecting Spam in Online Reviews," in *International Conference on Information and Communication Technology for Sustainable Development*, Dhaka, Bangladesh, Feb. 2021, pp. 296–299, https://doi.org/10.1109/ICICT4SD50815.2021.9396881.

[19] I. Frias-Blanco, A. Verdecia-Cabrera, A. Ortiz-Diaz, and A. Carvalho, "Fast adaptive stacking of ensembles," in *31st Annual ACM Symposium on Applied Computing*, Pisa, Italy, Apr. 2016, pp. 929–934, https://doi.org/10.1145/2851613.2851655.

[20] M. Abd El-Kareem, A. Elshenawy, and F. Elrfaey, "Mail spam detection using stacking classification," *Journal of Al-Azhar University Engineering Sector*, vol. 12, no. 45, pp. 1242–1255, Oct. 2017, https://doi.org/10.21608/auej.2017.19151.

[21] S. Madichetty, "A stacked convolutional neural network for detecting the resource tweets during a disaster," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3927–3949, Jan. 2021, https://doi.org/10.1007/s11042-020-09873-8.

[22] M. Anwer, S. M. Khan, M. U. Farooq, and Waseemullah, "Attack Detection in IoT using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, Jun. 2021, https://doi.org/10.48084/etasr.4202.

[23] V. C. Ho, T. H. Nguyen, T. Q. Nguyen, and D. D. Nguyen, "Application of Neural Networks for the Estimation of the Shear Strength of Circular RC Columns," *Engineering, Technology & Applied Science Research*, vol. 12, no. 6, pp. 9409–9413, Dec. 2022, https://doi.org/10.48084/etasr.5245.

[24] H. Oh, "A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model," *IEEE Access*, vol. 9, pp. 144121–144128, 2021, https://doi.org/10.1109/ACCESS.2021.3121508.

[25] C. Zhao, Y. Xin, X. Li, Y. Yang, and Y. Chen, "A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data," *Applied Sciences*, vol. 10, no. 3, Jan. 2020, Art. no. 936, https://doi.org/10.3390/app10030936.

[26] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in Twitter spam detection using ensemble learning," *Computers & Security*, vol. 69, pp. 35–49, Aug. 2017, https://doi.org/10.1016/j.cose.2016.12.004.

[27] T. O. Omotehinwa and D. O. Oyewola, "Hyperparameter Optimization of Ensemble Models for Spam Email Detection," *Applied Sciences*, vol. 13, no. 3, Jan. 2023, Art. no. 1971, https://doi.org/10.3390/app13031971.

[28] K. Sahu, F. A. Alzahrani, R. K. Srivastava, and R. Kumar, "Evaluating the impact of prediction techniques: Software reliability perspective," *Computers, Materials and Continua*, vol. 67, no. 2, pp. 1471–1488, 2021, https://doi.org/10.32604/cmc.2021.014868.

[29] "2007 TREC Public Spam Corpus." [Online]. Available: https://plg.uwaterloo.ca/~gvcormac/treccorpus07/.

[30] "The Enron-Spam datasets." https://www2.aueb.gr/users/ion/data/enron-spam/.

[31] K. Sahu and R. K. Srivastava, "Needs and Importance of Reliability Prediction: An Industrial Perspective," *Information Sciences Letters*, vol. 9, no. 1, pp. 33–37, Mar. 2020, https://doi.org/10.18576/isl/090105.

[32] M. A. Haq, "Smotednn: A novel model for air pollution forecasting and aqi classification," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 1403–1425, 2022, https://doi.org/10.32604/cmc.2022.021968.

[33] M. A. Haq, M. A. R. Khan, and M. Alshehri, "Insider Threat Detection Based on NLP Word Embedding and Machine Learning," *Intelligent Automation and Soft Computing*, vol. 33, no. 1, pp. 619–635, 2022, https://doi.org/10.32604/iasc.2022.021430.

[34] M. Z. Gashti, "Detection of Spam Email by Combining Harmony Search Algorithm and Decision Tree," *Engineering, Technology & Applied*

Science Research, vol. 7, no. 3, pp. 1713–1718, Jun. 2017, https://doi.org/10.48084/etasr.1171.

[35] M. Madhukar and S. Verma, "Hybrid Semantic Analysis of Tweets: A Case Study of Tweets on Girl-Child in India," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2014–2016, Oct. 2017, https://doi.org/10.48084/etasr.1246.

[36] M. A. Haq, M. A. R. Khan, and T. AL-Harbi, "Development of pccnn-based network intrusion detection system for edge computing," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 1769–1788, 2022, https://doi.org/10.32604/cmc.2022.018708.

[37] M. A. Haq, "DBoTPM: A Deep Neural Network-Based Botnet Prediction Model," *Electronics*, vol. 12, no. 5, Jan. 2023, Art. no. 1159, https://doi.org/10.3390/electronics12051159.

[38] M. A. Haq and M. A. R. Khan, "Dnnbot: Deep neural network-based botnet detection and classification," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 1729–1750, 2022, https://doi.org/10.32604/cmc.2022.020938.

[39] M. A. Haq, "CDLSTM: A novel model for climate change forecasting," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 2363–2381, 2022, https://doi.org/10.32604/cmc.2022.023059.

[40] M. A. Haq, A. K. Jilani, and P. Prabu, "Deep learning based modeling of groundwater storage change," *Computers, Materials and Continua*, vol. 70, no. 3, pp. 4599–4617, 2022, https://doi.org/10.32604/cmc.2022.020495.

[41] M. A. Haq *et al.*, "Analysis of environmental factors using AI and ML methods," *Scientific Reports*, vol. 12, no. 1, Aug. 2022, Art. no. 13267, https://doi.org/10.1038/s41598-022-16665-7.