Explainable AI-based Framework for Efficient Detection of Spam from Text using an Enhanced Ensemble Technique

Ahmed Alzahrani
Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
aaalzahrani9@kau.edu.sa (corresponding author)

Received: 21 May 2024 | Revised: 30 May 2024 | Accepted: 6 June 2024

ABSTRACT

Today, identifying and preventing spam has become a challenge, particularly with the abundance of text-based content in emails, social media platforms, and websites. Although traditional spam filters are somewhat effective, they often struggle to keep up with new spam methods. The introduction of Machine Learning (ML) and Deep Learning (DL) models has greatly improved the capabilities of spam detection systems. However, the black-box nature of these models poses challenges to user trust due to their lack of transparency. To address this issue, Explainable AI (XAI) has emerged, aiming to make AI decisions more understandable to humans. This study combines XAI with ensemble learning, utilizing multiple learning algorithms to improve performance, and proposes a robust and interpretable system to detect spam effectively. Four classifiers were used for training and testing: Support Vector Machine (SVM), Logistic Regression (LR), Gradient Boost (GB), and Decision Tree (DT). To reduce overfitting, two independent spam email datasets were blended and balanced. The stacking ensemble technique, based on Random Forest (RF), was the best-performing model compared to individual classifiers, having 98% recall, 96% precision, and 97% F1-score. By leveraging XAI's interpretability, the model elucidates the reasoning behind its classifications, leading to the comprehension of hidden patterns associated with spam detection.

Keywords-spam prediction; ensemble techniques; stacking classifier; explainable AI; machine learning; weak learner; strong learner

I. INTRODUCTION

The widespread expansion of the Internet has led to an increase in content, including a notable amount of web spam. Web spam, which includes practices such as manipulative content, diminishes the overall quality of online information, disrupts search engine results, and affects user satisfaction. This unwanted material, which includes ads and harmful spam scams, presents an obstacle to maintaining the safety and trustworthiness of online platforms [1]. Today, ML and DL have become valuable tools in detecting and filtering out web spam. These advanced technologies have shown effectiveness in identifying patterns and irregularities that suggest spam content. However, a significant challenge persists, as there is a lack of transparency in how they make decisions. As AI models become more sophisticated, their decision-making processes become less understandable, creating a dilemma known as the black-box issue that undermines user confidence and complicates adherence to regulations. Since lack of transparency creates problems for the web spam detection sector, it is very important to understand the reasons behind classification, refinement, and false positives to provide clear information to consumers and regulators [2].

A. *Research Motivation and Context*

Conventional spam detection systems mainly rely on set rules and basic statistics to adopt advanced ML algorithms that can analyze large amounts of data and detect subtle spam patterns. However, the increasing complexity of these models increases the lack of transparency in how they make decisions. XAI seeks to address this issue by ensuring that the results of the AI models are clear and easy to comprehend and interpret. Incorporating XAI into systems that detect web spam provides stakeholders with an understanding of why certain classifications are made, building trust, and facilitating model adjustments. However, blending explainability into spam detection models without compromising their performance presents a complex challenge [2].

B. *Problem Statement*

This study primarily addresses the problem of classifying spam from text using an ensemble learning technique. The purpose is to develop a predictive model that provides a class label $S_i \in \{0, 1\}$ in a text review r_i, where 1 indicates spam and 0 represents ham (non-spam) given a series of text reviews indicated as $\{r_1, r_2, r_3, \ldots, r_n\}$. The XAI module offers easily understandable rationales for its predictions.

C. Research Questions

RQ1: How to create an efficient ensemble method that combines various machine learning models to accurately distinguish between spam and ham in textual content?

RQ2: How effective are ensemble approaches compared to poor learners (classifiers)?

RQ: How does the proposed XAI-based ensemble approach for spam detection compare to previous baseline studies?

D. Research Contributions

This study expands [7], which used a spam dataset to develop a model that used Gradient Boosting (GB), Random Forest (RF), Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Support Vector Machine (SVM) to classify textual material into two classes: spam and ham. The proposed approach goes beyond, applying several classifiers via ensemble learning to classify textual input into binary classes. The proposed ensemble stacking classifiers use combined predictions from many models. In terms of accuracy scores, the proposed ensemble method outperforms state-of-the-art systems. Incorporating Shapley Additive exPlanations (SHAP) into the ensemble framework enhances the model's predictions by offering easily understandable explanations. This addresses the opacity of ML models, ensuring that users and stakeholders can efficiently comprehend and interpret the decision-making process.

II. RELATED WORK

In [6], K-Nearest Neighbors (KNN), SVM, RF, and NB were employed on a dataset with 47 features to classify emails into four groups: content, header, URL, and JavaScript. NB achieved 98% accuracy. In [7], a system was presented, based on feature extraction techniques such as TF-IDF. A spam SMS dataset was used on GBM, RF, SVM, GNB, and LR. RF classifiers achieved an accuracy rate of 99%. In [8], ML algorithms and Bidirectional Encoder Representations from Transformers (BERT) were employed. Email text was processed by BERT and its output features were used to represent the data. In both datasets, LR achieved better results than other methods. In [9], a variety of ML classifiers was used to detect spam and ham email. Bi-Long Short-Term Memory (Bi-LSTM) achieved an accuracy of 98.5% and an F1-score of 96%. In [10], LSTM and CNN with the Glove algorithm were used for autonomous feature extraction. This study compared traditional ML and DL models on a spam dataset to discover the best results. Using a benchmark dataset of 5243 spam and 16872 ham SMS, CNN with the Glove method achieved 96.52% accuracy. In [11], a DL method was introduced, based on Bi-LSTM, which outperformed other classifiers in terms of accuracy, achieving 93.4% in the ExAIS_sms dataset and 98.6% in the UCI dataset. In [12], Bi-LSTM and BERT models were used to identify and categorize spam emails on the Enron dataset. The accuracy rates for BERT were 98.34% and 97.15%, respectively. In [13], multiple algorithms were used, such as SVM, NB, and LSTM, to generate CSV and label files. Comparison of LSTM, NB, and SVM revealed that LSTM had the highest accuracy in detecting spam emails, with accuracy ratings of 99.62, 97, and 98%, respectively. In [14], various combinations of four classifiers, namely GNB, Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), and DT, were used. Then, a voting classifier was used to produce more accurate predictions than using separate classifiers. In [15], ML stacking ensemble techniques were used to improve the accuracy of spam email classification. Five classifiers were trained and tested, namely LR, DT, KNN, GNB, and AdaBoost, the latter being the best-performing model. In [16], HELPHED was presented, based on ensemble learning using stacking and soft-voting, achieving an F1-score of 0.99 with the soft-voting method surpassing previous techniques.

III. METHODOLOGY

Figure 1 depicts the proposed methodology.

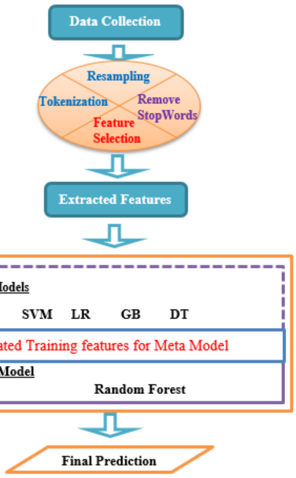

Fig. 1. Proposed system overview.

A. Dataset Acquisition

Table I shows details of the two datasets: D1 [17] and D2 [18]. The D1 dataset was split into 20% for testing and 80% for training. It contains 5573 reviews, 747 of which are classified as spam and 4825 as legitimate (non-spam). For the D2 dataset, this study omitted occurrences from the undecided group and focused only on binary classification. Table I details every category of items used during the experiments.

TABLE I. DATASET DETAILS

Dataset	Description	Numbers of reviews in labeled classes	
		SPAM	HAM
D1	Train	747 (13.40%)	4825 (86.57%)
D2	Train	208	3,641
	Test	113	1,835

B. Pre-processing and Data Preparation

The acquired dataset was pre-processed using the following steps.

1) Text Processing

Tokenization and removing stop words were used for text pre-processing using Python NLTK.

2) Re-Sampling Approach

Table I shows that there is skewness in the D2 dataset. Class imbalance is a common problem that arises when a model is trained on a dataset with a skewed or unequal distribution. Using random oversampling, the number of samples in the minority class was efficiently increased to the same level as in the majority class.

3) Training and Testing Data

70% of the overall dataset was used for training and 30% for testing.

4) Feature Engineering

Feature engineering included the following modules.

a) Count Vectorizer

A count vectorizer is used to tag entire text documents, encode new documents using pre-existing vocabulary, and create a term dictionary [6].

b) Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF is very useful in spam detection algorithms as it effectively balances the frequency of a term in a specific document. It is computed as:

$$Tf_{t,d} = \frac{\text{count of term in } d}{number\ of\ words\ in\ d} \quad (1)$$

$$IDF_{(t,D)} = \log(\frac{n}{DF+1}) \quad (2)$$

$$\mathrm{TF} - \mathrm{IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D) \quad (3)$$

C. Proposed Ensemble-Based Spam Recognition System

The model was designed to detect spam in text reviews, using text review and web spam datasets. A two-layer stacking ensemble was employed, as shown in Figure 2. The features of the original dataset include a label (spam or ham) and raw data that go through pre-processing. The features of the produced dataset are numerical representations obtained from the original text using Count Vectorizer and TF-IDF by considering word frequencies and their predictive significance. The ensemble model used feature selection techniques to generate meta-features from four base classifiers: DT, GB, LR, and SVM. These predictions were combined with the original dataset features to form a new dataset to train the RF meta-classifier. RF was selected as a meta-classifier since it is good at handling inconsistent data and reducing overfitting. A meta-model, such as RF, is used to improve overall performance, boost generalization, and reduce biases by combining likelihoods from base models.

1) Mathematical Formulation of Proposed Model

Stacking ensemble techniques combine the results of several base classification models to detect spam. This technique has the following mathematical model.

$S = \{(a_1, b_1), (a_2, b_2), \dots, (a_N, b_N)\}$ represents the training dataset, where a_i is the i^{th} example's feature vector and b_i is its matching spam/ham class. n, where $n = 4$, is the number of base classifiers, denoted by $g_1, g_2,\ g_3,\ g_4$ for DT, GB, LR, and SVM. The meta-classifier (RF) is denoted by g.

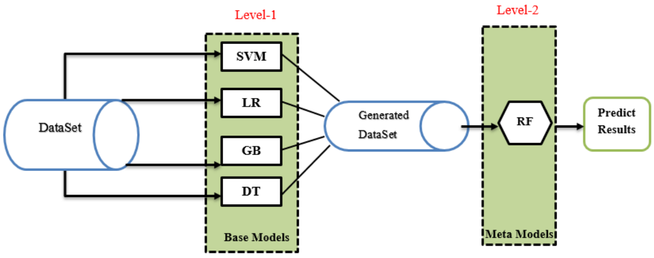

Fig. 2. Proposed 2-layer stacking ensemble method.

- Base classifiers training: For every base classifier $n = 1, 2, 3, 4$:

  $g_m \leftarrow train(S)$

  For each case i , obtain predictions from each base classifier:

  $h_{n,i} = g_m(a_i)\ for\ n = 1, 2, 3, 4$

- Development of meta-features: Make a new dataset S' and include the original features a_i and the predictions from each basic classifier for each sample.

  $(a_{i,} k_{1,i}, k_{2,i}, k_{3,i}, k_{4,i}, y_i)$

- Meta-classifier training (RF): $q \leftarrow train(S')$

  The RF meta-classifier is trained to acquire the ability to translate the predictions of the base classifier to the actual labels.

- Prediction stage: Using new input p, acquire predictions from each base classifier

  $h_m(p) = \ g_m(p) g\ or\ n = 1.2.3.4$

  Generate a new instance for the final prediction

  $\big(s, t_1(p), t_2(p), t_3(p), t_4(p)\big)$

- Employ the learned meta-classifier (RF): The final prediction $u(p)$ is obtained by:

  $u(p) = predict(u, \big(p, t_1(p), t_2(p), t_3(p), t_4(p)\big))$

D. Explanations of ML Models Using XAI

XAI helps bridge the gap between ML methods and understanding spam detection. It improves the identification of spam and non-spam by analyzing datasets, including texts, to detect patterns linked to spam. These complex models are often seen as black boxes, without understanding how they make their predictions. XAI aims to address this by making these models more transparent and understandable [27]. Developing SHAP to detect spam in text involving ensemble techniques requires customizing the calculation of SHAP values for this specific purpose. The goal is to dissect a model's prediction into contributions from features to gain insight into how it made a prediction based on input features, such as the text data obtained.

1) SHAP Value Calculation

SHAP in an XAI method to calculate important features [25]. Each likelihood calculated by the ensemble model can be described in terms of the contribution of each feature using SHAP, providing clear and in-depth insights into why a specific text is detected as spam or ham. SHAP values are calculated for distinct models within the ensemble, and these explanations are aggregated to provide complete details on the decision-making process. This approach not only increases understandability but also ensures that each result can be traced back to exact input features, thus increasing the reliability of the model. The SHAP value for a feature x_i in a prediction evaluates the extent to which x_i influences the difference between the model's prediction for input x and the average prediction of the model across all inputs. When examining a feature x_i, its SHAP value ϕ_i is computed as:

$$\varphi_i = \sum_{S \subseteq \mathrm{F}\{\mathrm{i}\}} \frac{|\mathrm{S}|!(|\mathrm{F}|- |\mathrm{S}|- 1)!}{|\mathrm{F}|!} [\mathrm{f}\ (\mathrm{S}\ \cup \{\mathrm{i}\}) - \ \mathrm{f}(\mathrm{S}) \qquad (5)$$

2) *Model Explanation*

The model's prediction, represented by p for a given instance, can be deconstructed as follows [26]:

$$p(x_i) = \varphi_{\mathrm{o}} + \sum_{i=0}^{n} \varphi_i\, \acute{x}_{\mathrm{i}} \qquad (6)$$

where φ_{o} represents the point that signifies the model's prediction, and $x_i{}'$ denotes the i^{th} feature value of the instance across the dataset. The total of SHAP values for all features $\sum_{i=0}^{n} \varphi_i$ explains the impact of each feature on deviating from the prediction value.

IV. RESULTS AND DISCUSSION

A. *Answer to RQ1*

Spam was predicted from the input text using ensemble learning, addressing RQ1. Its effectiveness as a well-structured heterogeneous ensemble approach led to its selection as the ensemble learning technique. Stacking functions as a super multi-layer model, where the models in the previous layer influence the results of the subsequent layer. Each layer in the stacking model consists of one or more models.

1) *Algorithmic Complexity of the Proposed System*

Assuming n data points and 4 base models, the complexity of the proposed ensemble-based spam prediction on the benchmark spam dataset using the stacking classifier can be calculated as follows:

- Training base classifiers: Total time is increased by the complexity of each base classifier. Let T_i represent the i^{th} base classifier's complexity. $\sum T_i$ would be the overall training complexity. Assume that the average training complexity of each base model is $O(n^2)$. As a result, the overall complexity of training the four base models is: $O(4n^2) = 4 \times O(n^2)$.
- Training stacking classifier: The stacking classifier learns how to combine the predictions from the underlying models to generate more accurate predictions. $O(m \times d)$ is the training efficiency of the stacking classifier, where m is the number of base models and d is the number of features. As a result, the following complexity results from training the stacking classifier with four base models $O(4d)$. The total level of complexity in the provided benchmark dataset of n data points and four base models is $O(4n^2) + O(4d)$. Since n is frequently much larger than d, the complexity's main term is $O(4n^2)$. Thus, the overall complexity of the ensemble approach can be calculated as $O(4n^2)$.

B. *Answer of RQ2*

Table II displays the results of the proposed model with its default parameter settings and weak classifiers. The proposed stacking classifier performed better than the weak classifiers in every metric, having an amazing accuracy of 99%.

TABLE II. RESULTS WITH DEFAULT PARAMETERS

Evaluation Metric	Class Label	Weak Learners				
		DT	GB	LR	SVM	Proposed
Precision (%)	Spam	0.93	0.85	0.82	0.79	0.96
	Ham	0.96	0.91	0.84	0.81	0.98
Recall (%)	Spam	0.92	0.87	0.84	0.83	0.98
	Ham	0.92	0.93	0.86	0.93	0.97
Accuracy (%)	Spam	0.94	0.86	0.81	0.81	0.97
	Ham	0.93	0.90	0.83	0.79	0.98
F1-Score (%)	Spam	0.95	0.85	0.83	0.80	0.97
	Ham	0.96	0.92	0.84	0.81	0.98

C. *Answer of RQ3*

The results in Table III show higher accuracy compared to baseline evaluations.

TABLE III. BASELINE METHODS AGAINST ENSEMBLE STACKING CLASSIFIER

Study	Aim	Technique	Dataset	Accuracy results (%)

[7]	Spam prediction	ML	Spam	Spam=90
				Ham=91

[14]	Spam classification - Ensemble method	Ensemble method	Spam	Spam=89
				Ham=88

Proposed	Spam vs Real text classification	Base models for ensemble stacking: SVM, LR, GB, DT. Meta-model: RF	Spam	Spam=97
				Ham=98

D. *Cross Validation*

The dataset was randomly partitioned into ten equal parts to perform 10-fold cross-validation. In each iteration, nine parts were merged into the training set, while the remaining was used as the test set. Each base learner provides a new feature indicating its performance under cross-validation. These attributes were then used to train the meta-learner, which effectively captures the capabilities of the base learners while preserving their anonymity [21-24]. Table IV reports key data.

TABLE IV. 10-FOLD CROSS-VALIDATION OF THE PROPOSED AND SINGLE MODELS

Model	Mean Accuracy	Accuracy SD	Mean Precision	Precision SD	Mean Recall
DT	92%	0.03	91%	0.05	91%
GB	88%	0.02	85%	0.04	89%
LR	82%	0.04	83%	0.05	86%
SVM	80%	0.03	84%	0.05	81%
Proposed	97.5%	0.02	96%	0.02	97%

E. Results Analysis for Explainable AI (XAI) Module

Figure 3 shows an analysis of SHAP results, merging information from the predicted category and individual features to offer a comprehensive overview.

TABLE V. ANALYSIS OF SHAP RESULTS

Step	Description	Example (SMS text: "Free gift! Click here https://www.shorturl.at")
Base value	Represents the average prediction of the ensemble model for the entire dataset, indicating the baseline spam probability.	Base value = 0.20 (20% chance of being spam)
Feature SHAP values	SHAP calculates the contribution of each feature (word) to the model's prediction for the specific SMS instance)	
	"Free" (SHAP value = 0.15)	"Free" increases the spam probability by 15% compared to the base value.
	Click (SHAP value = 0.20)	"Click" further pushed the prediction toward spam by 20%.
	URL (SHAP value = 0.10)	URL contributes to an additional 10% spam likelihood.
SHAP explanation	"Other words" (SHAP values = combined contribution of remaining words, might be negative for some)	
	Combines the base value and feature contributions to explain the final prediction.	SHAP explanation (0.20 + 0.15 + 0.20 + 0.10 + Other words) = 0.65 (65% being spam)
Feature interactions (optional)	SHAP can also consider interactions between features	
	"Free"×"Click" (SHAP interaction value)	Shows how the presence of both "Free" and "Click" influences the prediction compared to their individual effects.

F. Discussion

The ensemble method, enhanced by SHAP to provide clarity, showed an improvement in predicting accuracy compared to the baseline models. The accuracy achieved was 97% with precision, recall, and F1 scores of 96%, 98%, and 97%. The SHAP method helps to understand how the model makes decisions, showing which features matter most in identifying spam messages. Furthermore, using SHAP analysis helped uncover any biases that might exist in the model, guaranteeing fairness and dependability in spam detection. By clarifying how each feature influences the model's choices, SHAP gives confidence in the model results, allowing developers and stakeholders to improve the model using evidence.

G. Generalizability of Experimental Results

Spam Tactics: SHAP explanations would likely focus on outdated spam indicators if the model is not updated with fresh data.

Ensemble Configuration: The selection of ML models for the ensemble can affect the behavior and features emphasized by SHAP.

H. Feature Engineering

The SHAP explanations highlight the features utilized by the model. Their applicability relies on how pertinent they are for identifying spam in various situations.

I. Language:

SHAP explanations would capture distinct language features of the dataset being analyzed.

J. Integrating Domain Knowledge

By incorporating knowledge about spam trends and language nuances, feature engineering can be improved, and the effectiveness of SHAP explanations can also be enhanced.

K. Data Augmentation

Creating text messages that mimic spam techniques can assist the system and SHAP explanations adjust to changing trends.

V. CONCLUSION AND FUTURE WORK

This study focused on developing an effective XAI-based spam filtering technique that could distinguish between spam and ham text content. To address this issue, an ensemble technique was introduced, which used a stacking classifier approach to combine predictions from multiple models to improve accuracy, manage differences in model effectiveness, balance bias-variance trade-offs, and improve generalization by leveraging individual model strengths. This multi-layered stacking classifier uses the predictions from the first layer (base models) to train a meta-classifier in the second layer. The chosen base models were SVM, LR, BG, and DT classifiers, with an RF serving as the meta-model. The XAI module allowed the understanding of the rationale behind the model's predictions. This system evaluates text inputs to determine if they are spam or ham. The SHAP XAI module provided understandable reasons for the model's predictions, connecting the effectiveness of the system with its comprehensibility. These explanations not only enhance performance but also boost confidence in AI. The proposed stacking classifier was highly effective in classifying spam (97% accuracy) and ham (98% accuracy), indicating a significant improvement in spam detection capabilities. Furthermore, it gives useful insights to further improve spam detection methods. However, depending exclusively on text for spam identification can be limited. Therefore, including multimedia features, such as audio-visual information, photos, and emojis, can provide better spam detection.

Although DL models such as LSTM, CNN, BERT, and Bi-LSTM are available, the proposed study applied SVM, LR, GB, and DT classifiers due to their balance of simplicity, interpretability, and efficiency. These models work extremely efficiently in binary classification tasks. Compared to DL models, ML models use less computational power. ML models also work well with XAI methods such as SHAP. Due to their accuracy, implementation, and transparency, they are a suitable option for the research goals of this study. More research is needed to explore the social impact of implementing AI-driven spam detection on communication platforms. This includes mitigating bias, protecting privacy, and understanding how false negatives and positives can affect users. Partnering with ethicists, sociologists, and legal professionals would offer a rounded perspective on these matters.

REFERENCES

[1] A. Ibrahim, M. Mejri, and F. Jaafar, "An Explainable Artificial Intelligence Approach for a Trustworthy Spam Detection," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, Jul. 2023, pp. 160–167, https://doi.org/10.1109/CSR57506.2023.10224956.

[2] Z. Zhang, E. Damiani, H. A. Hamadi, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence to Detect Image Spam Using Convolutional Neural Network," in *2022 International Conference on Cyber Resilience (ICCR)*, Dubai, United Arab Emirates, Oct. 2022, pp. 1–5, https://doi.org/10.1109/ICCR56254.2022.9995839.

[3] Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104–93139, 2022, https://doi.org/10.1109/ACCESS.2022.3204051.

[4] T. Wu, S. Wen, Y. Xiang, and W. Zhou, "Twitter spam detection: Survey of new approaches and comparative study," *Computers & Security*, vol. 76, pp. 265–284, Jul. 2018, https://doi.org/10.1016/j.cose.2017.11.013.

[5] M. Z. Asghar, A. Ullah, S. Ahmad, and A. Khan, "Opinion spam detection framework using hybrid classification scheme," *Soft Computing*, vol. 24, no. 5, pp. 3475–3498, Mar. 2020, https://doi.org/10.1007/s00500-019-04107-y.

[6] H. Khan, M. U. Asghar, M. Z. Asghar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, "Fake Review Classification Using Supervised Machine Learning," in *Pattern Recognition. ICPR International Workshops and Challenges*, 2021, pp. 269–288, https://doi.org/10.1007/978-3-030-68799-1_19.

[7] M. A. Abid, S. Ullah, M. A. Siddique, M. F. Mushtaq, W. Aljedaani, and F. Rustam, "Spam SMS filtering based on text features and supervised machine learning techniques," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 39853–39871, Nov. 2022, https://doi.org/10.1007/s11042-022-12991-0.

[8] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 1, pp. 5–9, 2023, https://doi.org/10.47852/bonviewJCCE2202192.

[9] P. Malhotra and S. Malik, "Spam Email Detection Using Machine Learning and Deep Learning Techniques," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022*, 2022, https://doi.org/10.2139/ssrn.4145123.

[10] A. Sheneamer, "Comparison of Deep and Traditional Learning Methods for Email Spam Filtering," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021, https://doi.org/10.14569/IJACSA.2021.0120164.

[11] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli, "A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 17, 2022, Art. no. e6989, https://doi.org/10.1002/cpe.6989.

[12] K. Debnath and N. Kar, "Email Spam Detection using Deep Learning Approach," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, Faridabad, India, May 2022, vol. 1, pp. 37–41, https://doi.org/10.1109/COM-IT-CON54601.2022.9850588.

[13] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex & Intelligent Systems*, vol. 9, no. 3, pp. 3043–3070, Jun. 2023, https://doi.org/10.1007/s40747-022-00760-3.

[14] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, "Spam Detection Using Ensemble Learning," in *Harmony Search and Nature Inspired Optimization Algorithms*, 2019, pp. 661–668, https://doi.org/10.1007/978-981-13-0761-4_63.

[15] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, "Improving spam email classification accuracy using ensemble techniques: a stacking approach," *International Journal of Information Security*, vol. 23, no. 1, pp. 505–517, Feb. 2024, https://doi.org/10.1007/s10207-023-00756-1.

[16] P. Bountakas and C. Xenakis, "HELPHED: Hybrid Ensemble Learning PHishing Email Detection," *Journal of Network and Computer Applications*, vol. 210, Jan. 2023, Art. no. 103545, https://doi.org/10.1016/j.jnca.2022.103545.

[17] "SMS Spam Collection Dataset." [Online]. Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset.

[18] "Webspam-UK2007". [Online]. Available: https://chato.cl/webspam/datasets/uk2007/

[19] A. S. Khan, H. Ahmad, M. Zubair, F. Khan, A. Arif, and H. Ali, "Personality Classification from Online Text using Machine Learning Approach," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 460–476, 2020, https://doi.org/10.14569/IJACSA.2020.0110358.

[20] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security*, vol. 110, Nov. 2021, Art. no. 102414, https://doi.org/10.1016/j.cose.2021.102414.

[21] M. Z. Asghar, A. Khan, S. R. Zahra, S. Ahmad, and F. M. Kundi, "Aspect-based opinion mining framework using heuristic patterns," *Cluster Computing*, vol. 22, no. 3, pp. 7181–7199, May 2019, https://doi.org/10.1007/s10586-017-1096-9.

[22] U. A. Mohammed and M. Sanusi, "An Optimized Phising Email Detection and Prevention Using Classification Models," *International Journal of Engineering Applied Sciences and Technology*, vol. 7, no. 10, pp. 9–21, Feb. 2023, https://doi.org/10.33564/IJEAST.2023.v07i10.002.

[23] A. Alzahrani and M. Z. Asghar, "Cyber vulnerabilities detection system in logistics-based IoT data exchange," *Egyptian Informatics Journal*, vol. 25, Mar. 2024, Art. no. 100448, https://doi.org/10.1016/j.eij.2024.100448.

[24] A. Alzahrani, "Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13671–13680, Apr. 2024, https://doi.org/10.48084/etasr.7029.

[25] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017, https://doi.org/10.1109/ACCESS.2017.2747560.

[26] K. Roshan and A. Zafar, "Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation(SHAP)," *International journal of Computer Networks & Communications*, vol. 13, no. 6, pp. 109–128, Sep. 2021, https://doi.org/10.5121/ijcnc.2021.13607.

[27] K. Roshan and A. Zafar, "Using Kernel SHAP XAI Method to Optimize the Network Anomaly Detection Model," in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, Mar. 2022, pp. 74–80, https://doi.org/10.23919/INDIACom54597.2022.9763241.