

Comparative Analysis of Machine Learning Classifiers for E-mail Spam Filtering with RFE-SVM Feature Selection and SA-Optimised XGBoost

Abubakar HASSAN^{1*}, Emmanuel G. DADA²

^{1*}Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria

²Department of Computer Science, University of Maiduguri, Maiduguri, Nigeria

^{1*}abhassan@abu.edu.ng, ²gbengadada@unimaid.edu.ng

Abstract

This study presents a comparative evaluation of several machine learning classifiers for email spam filtering using a Kaggle dataset of 5,172 email samples. The investigation includes a proposed hybrid approach combining RFE-SVM feature selection with SA-optimized XGBoost, evaluated alongside SVM, MLP, Gaussian Naïve Bayes, and Logistic Regression. Results showed that MLP achieved the highest accuracy at 99.2%, followed by SVM at 99.0%, while the proposed SA-XGBoost achieved 98.3% with notably lower memory consumption of 0.20 MB. The findings suggest that for this dataset, simpler classifiers may outperform more complex ensemble methods, and that the proposed approach offers a favourable accuracy-memory trade-off for resource-constrained environments.

Keywords: E-mail spam filtering, machine learning, XGBoost, Simulated Annealing optimization, RFE-SVM feature selection.

1.0 Introduction

Spam email message also referred to as junk or unsolicited emails [1]. They pose a great threat by consuming users' time and system resources, and potentially leaking confidential messages [2]. The rapid rise in email communication has been accompanied by a growth in spam, which overwhelms inboxes, spreads viruses, and present huge cybersecurity threats [3]. Not only slow productivity, but unsolicited email messages also serve as vectors for phishing, malware, and economic fraud. Despite all the technological advances in filtering solutions, fraudsters continue to evolve their tactics, making conventional detection techniques less effective [4].

Statistics show that emails account for up to 90% of all cyberattacks. Email communications platforms are characterized by vulnerabilities which cybercriminals exploit frequently. Spam and phishing emails remain the most prevalent categories of cybersecurity threat. Therefore, it is necessary to solve these devastating email-based cybersecurity risks and strengthen network security measures to minimize potential risks [1].

The number of email spam messages continues to increase at a rapid rate. Detecting and classifying these email spam messages prove difficult and present great challenge for email communication platforms [5]. Conventional techniques for email spam filtering such as blocklists, real-time blackhole lists, and content-based approaches suffer from notable shortcomings [6], [7]. Therefore, there has been a shift to more advanced computational intelligence approaches such as machine learning (ML) and deep learning (DL) to improve the accuracy of email spam detection. ML and DL techniques have recently achieved remarkable success for their effectiveness in enhancing spam filtering performance [8], [9]. ML has become a robust tool for spam filtering. While some single classifiers can perform well, ensemble methods offer advantages in certain scenarios [10], [11].

Ensemble learning has emerged as a powerful technique [12]. It employs the collective decision-making of multiple models to enhance accuracy, robustness, and adaptability [13]. Ensemble techniques prevent against the biases and variances naturally embedded in individual classifiers by combining their predictions through mechanism such as voting, stacking or boosting [14].

In this paper, we investigated the effectiveness of an integrated approach combining feature selection, ensemble learning, and model optimization against standard classifiers in terms of accuracy, computational efficiency, and memory usage. Specifically, Recursive Feature Elimination with SVM (RFE-SVM) was utilized to enhance feature selection by identifying the most relevant attributes. Meanwhile, the SA algorithm optimized the hyperparameters of XGBoost, resulting in a hybrid architecture.

1.1 Related Works

In this section, studies that have used ensemble learning for email spam detection are discussed. Adnan et al. [14] developed an ensemble learning technique using stacking strategy to improve classification accuracy of

email spam filtering. The authors trained and tested five individual machine learning models (LR, DT, KNN, Gaussian NB, and AB) on a balanced aggregation of two spam email datasets. Experimental results show that AB achieved superior performance compared to other classifiers. Stacking approach was also introduced which combines the predictions of all the base models employed. The stacking ensemble learning technique obtained superior performance with the highest accuracy (98.8%), recall (98.8%), and F1 score (98.9%) in comparison to the individual models.

Rayan [7] developed an ensemble learning using hybrid bagging technique for email spam filtering. The researchers employed DT and RF classifiers to enhance classification accuracy. In the preprocessing phase, they employed tokenization, stemming, and stop word removal. Correlation feature selection CFS is used to select the necessary features from the preprocessed data. The technique obtained an accuracy of 98%.

Omotehinwa and Oyewola[6] proposed ensemble learning technique for enhanced spam email detection using RF and Extreme Gradient Boost (XGBoost). The authors developed baseline classifiers using the Enron dataset. The base models were then optimized using grid-search cross-validation. Experimental results show that hyperparameter tuning significantly improved the performance of base classifiers. The Optimized XGBoost model achieved a higher accuracy, sensitivity, and F1 score compared to the optimized RF model.

Bountakas and Xenakis [15] proposed HELPHED, a technique to filter phishing emails using ensemble learning integrated with hybrid features. The stacking ensemble learning strategy employs DT and KNN as base models, with a Multilayer Perceptron (MLP) as a fusion model. The soft voting ensemble combines DT and KNN predictions using probability averaging. The technique was tested on a large, imbalanced dataset. Experimental results demonstrate that soft voting ensemble Learning outperforms other ML and DL techniques. It achieved an F1-score of 0.9942, while the other models achieved high performance.

In Hadi and Baawi's recent research [16], a stacking ensemble learning with recursive feature elimination technique is used for email spam detection. The stacking ensemble learning architecture combines XGBoost, RF, and Multinomial NB. They achieved an accuracy of 99.28% accuracy rate. The findings illustrate how feature-aware ensemble methods can greatly improve the performance and efficiency of email spam detectors.

Existing works have not systematically investigated the combination of RFE-SVM feature selection with SA-optimized XGBoost on the Kaggle dataset used, and that this study provides a comprehensive comparative evaluation.

2.0 Materials and Methods

2.1 Dataset Description

The email spam dataset used for this study was obtained from Kaggle. It is a CSV file containing related information of 5172 randomly picked email files and their respective labels for spam or not-spam classification. The csv file contains 5172 rows, each row for each email. There are 3002 columns. The first column indicates Email name. The name has been set with numbers and not recipients' name to protect privacy. The last column has the labels for prediction: 1 for spam, 0 for not spam. The remaining 3000 columns are the 3000 most common words in all the emails, after excluding the non-alphabetical characters/words. For each row, the count of each word(column) in that email(row) is stored in the respective cells. Thus, information regarding all 5172 emails are stored in a compact data frame rather than as separate text files (<https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv>).

2.2 Preprocessing

The text content of each email message undergoes preprocessing to have a high-quality input for the classification process. This starts with converting all text to lowercase to eliminate case sensitivity and ensure uniformity. Punctuation and special characters are removed to reduce noise. This is followed by the elimination of stop-words such as "the," "and," or "is," which carry little semantic value in distinguishing between spam and ham. The cleaned text is then tokenized into individual words (tokens), which serve as the basic units for feature extraction. The output of this phase is a structured and tokenized version of each email message.

2.3 Feature Engineering using RFE with SVM

The integration of SVM and RFE supports in identifying and choosing relevant features for building an effective predictive model. RFE iteratively removes the least important features based on the SVM weights until reaching the optimal number of features [17].

The SVM weight indicates how much a feature contributes to the overall model. A feature that has a higher SVM weight has a greater influence on the decision boundary of the model. On the other hand, the RFE rank evaluates how effectively the model functions in the absence of the feature. The dataset is divided into training and testing subsets, with a random allocation following a specific ratio of 4:1, where 80% of the data is designated for training purposes and the remaining 20% is earmarked for testing the classifier's performance. For feature selection, the training data (80%) with all variables is used as input for the feature selection process,

which identifies relevant features for building an effective predictive model. The training samples with selected features are then used to train the XGBoost model with 5-fold cross-validation. Each fold optimizes the XGBoost hyperparameters using the SA algorithm. Finally, the optimal hyperparameters and selected features from the SVM-based RFE process are applied in the testing phase. Here, the XGBoost classifier, using its tuned parameters, aims to maximize classification accuracy.

2.4 XGBoost Classifier

The XGBoost algorithm is an ensemble learning technique based on gradient boosting. It has advantages of fast computation, robustness, and accurate prediction in imbalanced dataset [17]. It also effectively addresses the problem of overfitting through regularization objective function and second-order Taylor expansion of the loss function. Furthermore, it incorporates both L1 (LASSO) and L2 (Ridge) regularization into its objective function, penalizing complexity and encouraging a simpler, more generalized model [17].

2.5 Hyperparameter Optimization with Simulated Annealing

Like other ML and ensemble classifiers, optimizing the hyperparameters of XGBoost is key for enhancing its accuracy and predictive performance. XGBoost has an array of hyperparameters that can be finetuned that plays a distinct role in controlling the complexity and generalization capabilities of the classifier [1]. In this study, Simulated Annealing (SA) was employed to finetune the hyperparameters of the XGBoost classifier.

SA is a heuristic optimization algorithm used to arrange components to minimize a cost function like wire length, delay, or area. It has been widely employed in VLSI design, image processing, computer vision, and neural network research [18], [19]. The algorithm starts with an initial placement and iteratively adjusts the positions of components using random movement. At each step, changes that lower the cost are accepted, while worse changes are accepted with a probability that decreases as the temperature parameter lowers. The algorithm converges to a near-optimal placement [20]. The proposed methodology is illustrated in Figure 1.

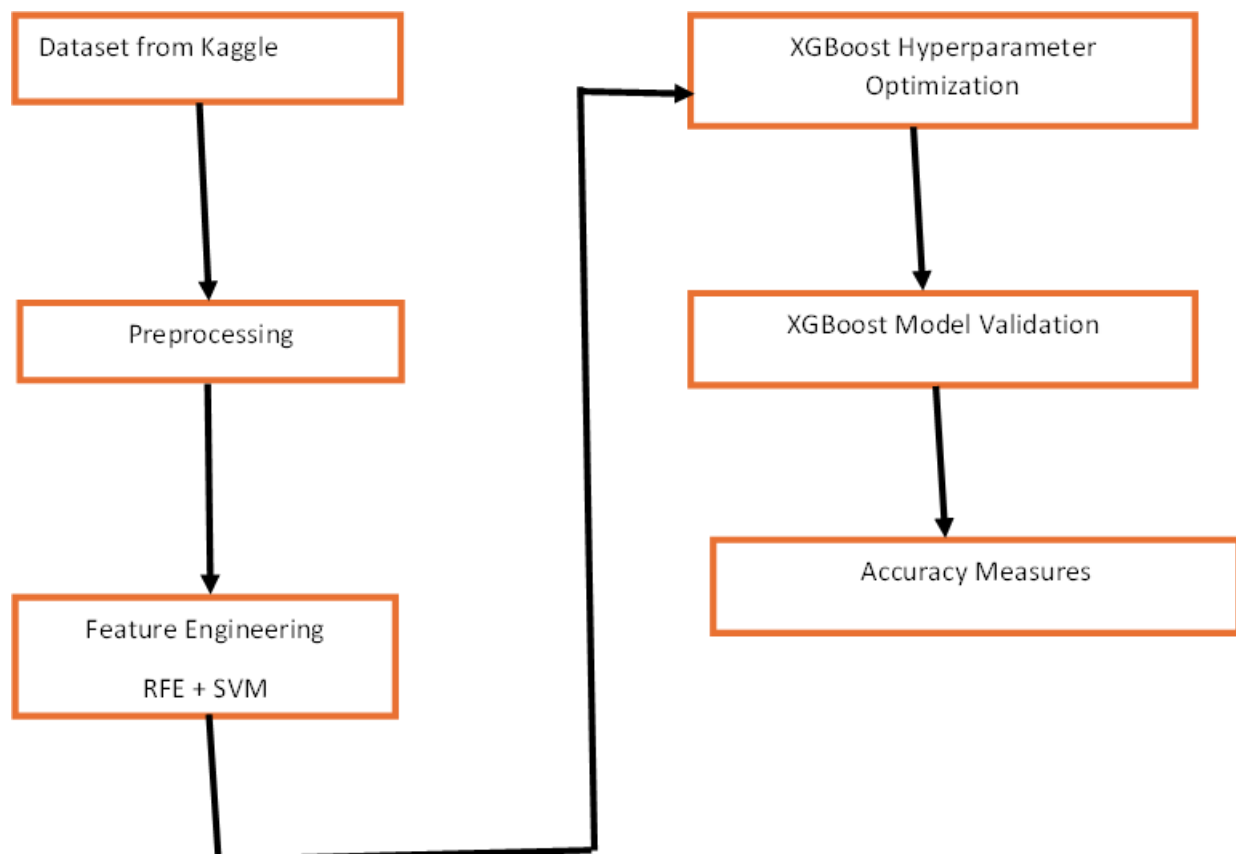


Figure 1: The Proposed SVM-RFE, XGBoost and Simulated Annealing Optimization Approach

2.6 Performance Evaluation

Confusion matrix is a tool used to evaluate the performance of a classification model. In this matrix, the rows represent the actual classes of the samples, while the columns represent the predicted classes. By employing the insights derived from a confusion matrix, one can effectively identify the strengths of the classifier while also recognizing areas that require improvement. The performance metrics utilized are

represented by the equations given below. Accuracy is a crucial metric that represents the total number of samples that have been correctly predicted by the model. It serves as a key indicator of performance and reliability of a model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall is the sensitivity of the model, and it is the ratio of the items that the model correctly classified as positive (TP) out of the correctly classified as positive and incorrectly classified as negative (TP + FN).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4)$$

Brier-score measures the accuracy of the probabilistic predictions and is defined as follows:

$$Brier - score = \frac{1}{N} \sum_{i=1}^N (p_i - e_i)^2 \quad (5)$$

where p_i is the predicted probability, e_i is the expected probability and N is the number of predicted instances.

2.7 Experimental and Parameter Setting

The experimental and parameter settings for the study are shown in Table 1. The models were trained with the pandas, NumPy and scikit-learn tools for numerical computation in Python programming environment.

Table 1: Experimental Settings and Parameter Tuning.

Model	Hyperparameter	Values		
SA-XGBoost + RFE-SVM	random_state	42		
XGBoost + RFE-SVM	random_state	42		
SVM + RFE-SVM	random_state	42		
LR + RFE-SVM	max_iter	2000		
MLP +RFE-SVM	hidden_layer_sizes.	100	500	42
	max_iter			
	random_state			
GNB+RFE-SVM				

3.0 Results and Discussion

This section presents the results and examines the significant discoveries derived from the simulations. The experiment was carried out in Python 3(jupyter) environment.

3.1 Confusion Matrix

A confusion matrix is a table which provides the performance summary of a machine learning classification algorithm by comparing its predicted labels to the true labels. It shows the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) of the classifier's predictions. The confusion matrices of XGBoost, SA-XGBoost, SVM, LR, MLP, and GNB are depicted in Figures (2- 7) respectively. SA-XGBoost performed better than XGBoost, LR, and GNB. While SVM and MLP performed slightly better than SA-XGBoost.

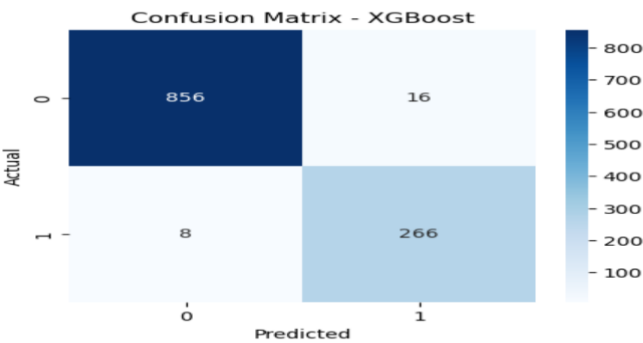


Figure 2: Confusion Matrix of XGBoost

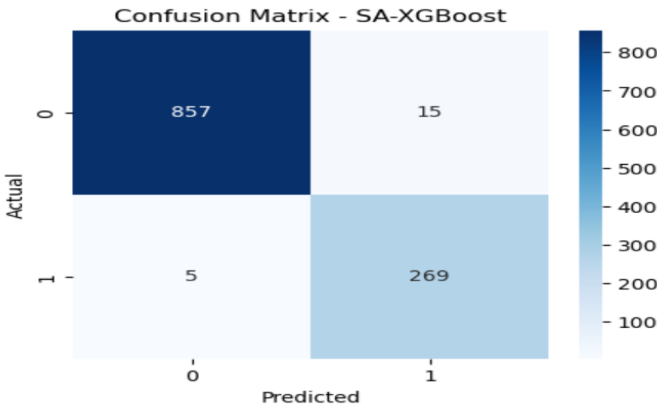


Figure 3: Confusion Matrix of SA-XGBoost

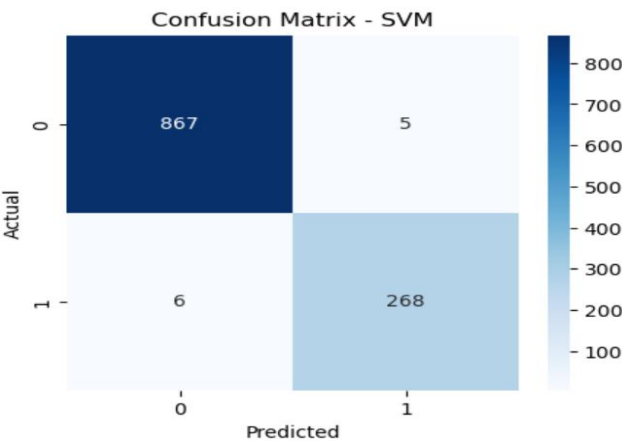


Figure 4: Confusion Matrix of SVM

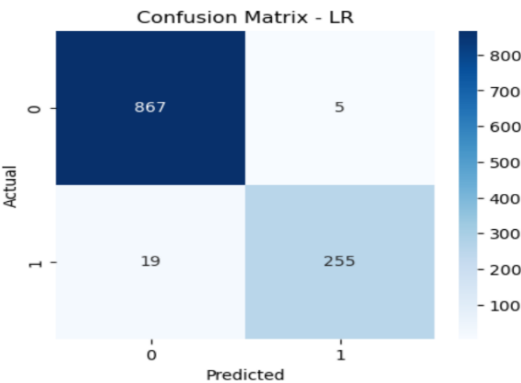


Figure 5: Confusion Matrix of LR

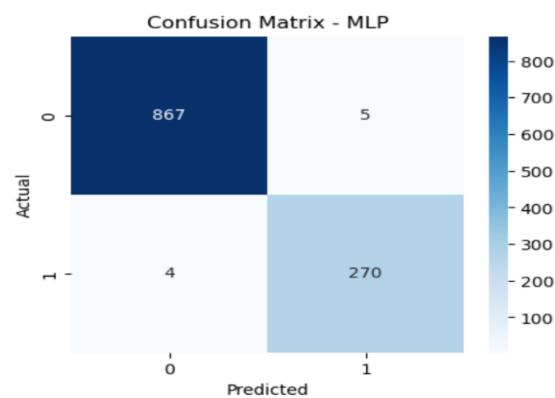


Figure 6: Confusion Matrix of MLP

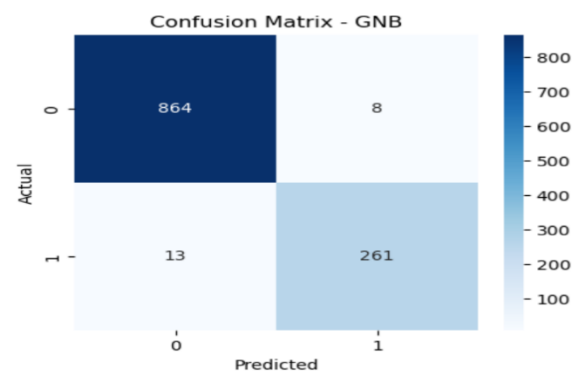


Figure 7: Confusion Matrix of GNB

3.2 ROC Curves

The performance analysis of the classifiers is also illustrated using ROC curves to provide insight into the trade-offs between sensitivity (recall) and specificity. It plots True Positive Rate (Recall) against the False Positive Rate (1-Specificity). The Area Under the Curve (AUC) indicates the extent of separability and measures how good a model is at classifying between positive and negative classes. Figures (8-13) illustrate the ROC of all the models. The AUC values fall within the interval between 0 and 1. The closer the value is to one, the more intelligent the model. All models obtained good scores results which indicates their great prediction performance.

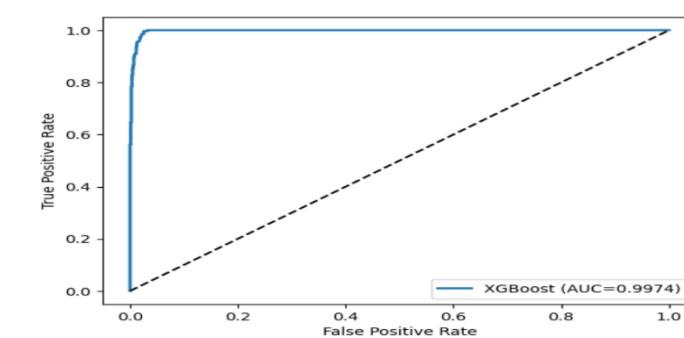


Figure 8: ROC Curve of XGBoost

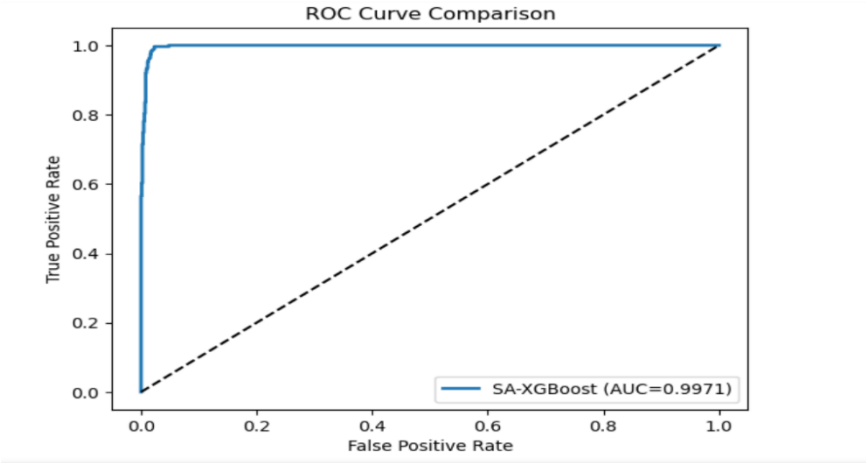


Figure 9: ROC Curve of SA-XGBoost

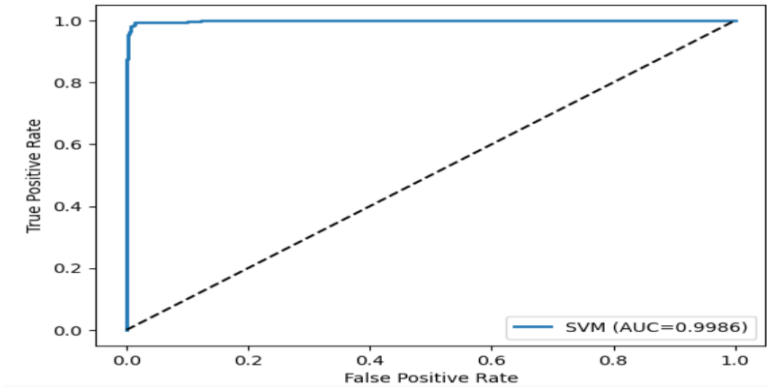


Figure 10: ROC Curve of SVM

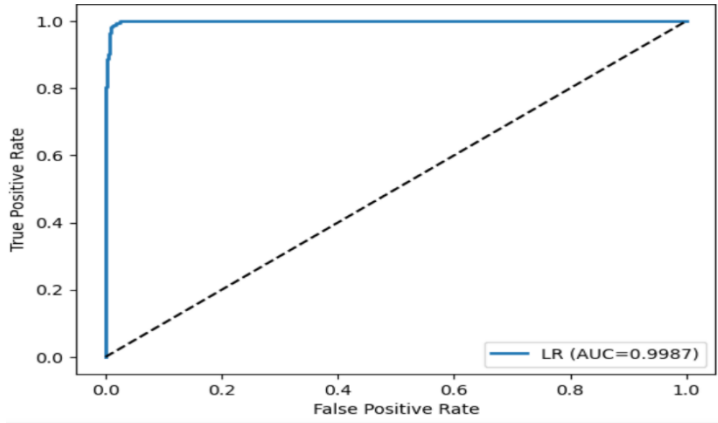


Figure 11: ROC Curve of LR

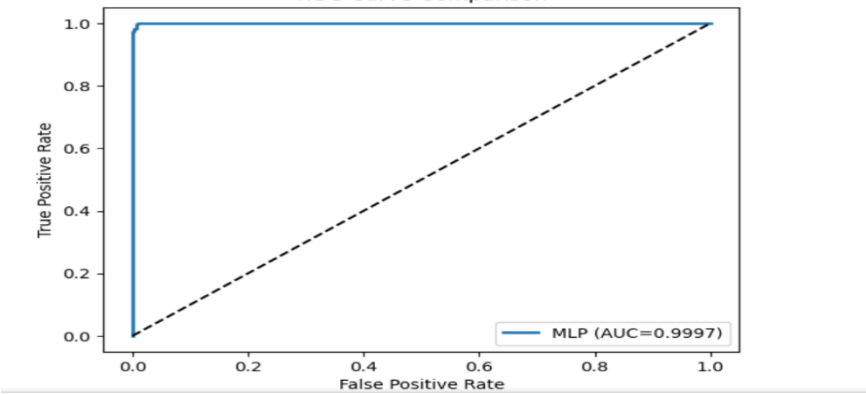


Figure 12: ROC Curve of MLP

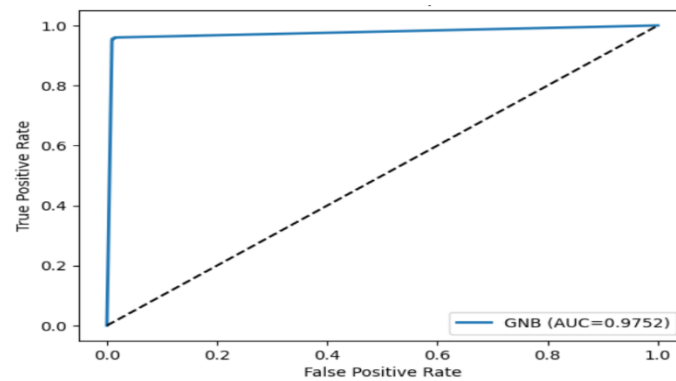


Figure 13: ROC curve of GNB

3.3 Models' Performance Comparison

Table 2 illustrates the performance comparison of models used in the study. SA-XGBoost achieved better accuracy results compared to XGBoost, LR and GNB models. While MLP obtained the highest results followed by SVM classifier. It can also be seen that SA-XGBoost and XGBoost have least memory consumption of 0.20 MB. Figure 14 shows the models' performance comparison.

Table 2: Models' Performance Comparison

Model	Accuracy	Precision	Recall	F1	AUC	Brier-Score	Time Per Sample(sec)	Memory (MB)
MLP	0.992	0.982	0.985	0.984	0.999	0.005	1.65e-05	6.06
SVM	0.990	0.982	0.978	0.980	0.998	0.009	7.33e-04	8.70
SA-XGBoost	0.983	0.947	0.964	0.964	0.997	0.015	1.46e-05	0.20
GNB	0.981	0.970	0.961	0.961	0.975	0.018	1.24e-04	88.44
XGBoost	0.979	0.943	0.957	0.957	0.997	0.016	0	0.20
LR	0.979	0.981	0.931	0.956	0.998	0.027	8.544e-07	0.62

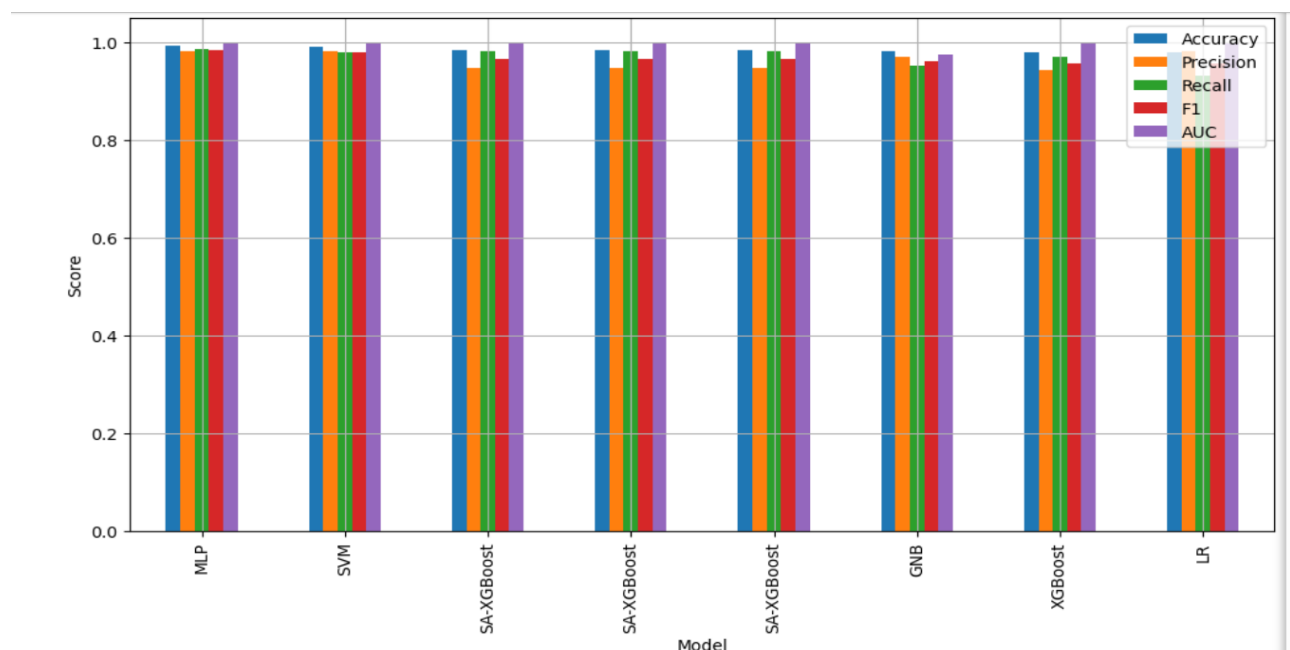


Figure 14: Models' Performance Comparison

4.0 Conclusion

In this study, we investigated the effectiveness of an integrated approach combining feature selection, ensemble learning, and model optimization. Specifically, Recursive Feature Elimination with SVM (RFE-SVM) was utilized to enhance feature selection by identifying the most relevant attributes. Meanwhile, the SA algorithm optimized the hyperparameters of XGBoost, resulting in an accurate and robust hybrid architecture. Simulation results demonstrated that SVM and MLP achieved higher accuracy but SA-XGBoost offers competitive accuracy with lower memory consumption. In future work, the proposed approach will be tested on larger, more diverse datasets.

References

- [1] S. Alam, A. Jameel, Z. Parveen, and E. Alnaway, "SHRED: An Ensemble-Based Machine Learning Model to Sift Email Messages for Real-Time Spam Detection," *IEEE Access*, vol. 13, pp. 157276–157293, 2025, doi: 10.1109/ACCESS.2025.3605850.
- [2] A. Hassan and M. A. Wajiro, "BES-SVM: Enhanced Machine Learning Algorithm for Telegram Spam Classification," *FUOYE Journal of Engineering and Technology*, vol. 10, no. 4, Dec. 2025, doi: 10.4314/fuoyej.v10i4.2.
- [3] A. Hassan, Y. Ayuba, M. A. Wajiro, and M. Z. Ahmad, "Machine Learning Algorithms For Telegram Spam Filtering," *FUDMA Journal of Sciences*, vol. 8, no. 6, pp. 170–176, Dec. 2024, doi: 10.33003/fjs-2024-0806-2799.
- [4] A. Hassan, M. Abatcha, and E. G. Dada, "Ensemble Machine Learning Algorithm for Telegram Spam Detection," *Arid-zone Journal of Basic and Applied Research*, vol. 3, no. a4, pp. 87–95, Aug. 2024, doi: 10.55639/607.06050402.
- [5] B. Majumder, A. Chowdhury, A. M. Hasan, D. I. Didar, B. Islam, and S. Maruf, "A Multi-View Feature Fusion approach with XLM-Roberta for Spam Email Detection," in *2025 28th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar: IEEE, Dec. 2025.
- [6] T. O. Omotehinwa and D. O. Oyewola, "Hyperparameter Optimization of Ensemble Models for Spam Email Detection," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031971.
- [7] A. Rayan, "Analysis of e-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/2500772.
- [8] E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, "Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems," *IEEE Access*, vol. 12, pp. 143627–143657, 2024, doi: 10.1109/ACCESS.2024.3467996.
- [9] S. Sarker, R. Bhattacharjee, and S. Adrita "Email Spam Detection Using Logistic Regression and Explainable AI," in *2025 International Conference on Electrical, Computer and Communication Engineering, ECCE 2025*, Institute of Electrical and Electronics Engineers Inc., 2025. doi: 10.1109/ECCE64574.2025.11013300.
- [10] N. Al-Shanableh Mazen, S. Alzyoud, and E. Nashnush, "Enhancing Email Spam Detection Through Ensemble Machine Learning: A Comprehensive Evaluation of Model Integration And Performance Model Integration and Performance," *Conference Proceedings 2024*, 2024, [Online]. Available: <https://scholarworks.lib.csusb.edu/ciima>
- [11] N. Arfaoui, "Enhancing Machine Learning Algorithms for Imbalanced Data—A Case Study: Spam Detection," *IEEE Access*, vol. 14, pp. 16070–16088, 2026, doi: 10.1109/ACCESS.2026.3658624.
- [12] B. A. Omodunbi and H. A. Olasunkanmi, "Development of an Ensemble Model for Email Filtering and Classification," *FUDMA Journal of Engineering and Technology*, 2026, [Online]. Available: <https://fjet.fudutsinma.edu.ng/index.php/fjet/>
- [13] S. Sarker, M. Rahman, and M. Talha, "Hybrid Email Filtering Using TF-IDF and BERT Enhanced Spam Detector," in *2025 2nd International Conference on Next-Generation Computing, IoT and Machine Learning, NCIM 2025*, Institute of Electrical and Electronics Engineers Inc., 2025. doi: 10.1109/NCIM65934.2025.11160311.
- [14] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, "Improving spam email classification accuracy using ensemble techniques: a stacking approach," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 505–517, Feb. 2024, doi: 10.1007/s10207-023-00756-1.
- [15] P. Bountakas and C. Xenakis, "HELPHED: Hybrid Ensemble Learning Phishing Email Detection", [Online]. Available: <https://ssrn.com/abstract=4147334>
- [16] M. T. Hadi and S. S. Baawi, "Stacking Ensemble Learning with Recursive Feature Elimination Technique for Email Spam Detection," in *AIP Conference Proceedings*, American Institute of Physics, Mar. 2025. doi: 10.1063/5.0260001.
- [17] S. Ayad, H. A. Al-Jamimi, and A. El Kheir, "Integrating Advanced Techniques: RFE-SVM Feature Engineering and Nelder-Mead Optimized XGBoost for Accurate Lung Cancer Prediction," *IEEE Access*, vol. 13, pp. 29589–29600, 2025, doi: 10.1109/ACCESS.2025.3536034.

- [18] V. F. Yu, H. Susanto, P. Jodiawan, T. W. Ho, S. W. Lin, and Y. T. Huang, "A Simulated Annealing Algorithm for the Vehicle Routing Problem With Parcel Lockers," *IEEE Access*, vol. 10, pp. 20764–20782, 2022, doi: 10.1109/ACCESS.2022.3152062.
- [19] Z. Elgamal, A. Q. M. Sabri, M. Tubishat, D. Tbaishat, S. N. Makhadmeh, and O. A. Alomari, "Improved Reptile Search Optimization Algorithm Using Chaotic Map and Simulated Annealing for Feature Selection in Medical Field," *IEEE Access*, vol. 10, pp. 51428–51446, 2022, doi: 10.1109/ACCESS.2022.3174854.
- [20] F. He and Q. Ye, "A Bearing Fault Diagnosis Method Based on Wavelet Packet Transform and Convolutional Neural Network Optimized by Simulated Annealing Algorithm," *Sensors*, vol. 22, no. 4, Feb. 2022, doi: 10.3390/s22041410.