Jurnal CoreIT, Vol.11, No.1, June 2025
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

SMS Phishing Detection Model with Hyperparameter Optimization in Machine Learning

Rahmad Abdillah[1], Fitri Insani[1*]
[1]Dept. of Informatics Engineering, Universitas Islam Negeri Sultan Syarif Kasim, Indonesia
rahmad.abdillah@uin-suska.ac.id, fitri.insani@uin-suska.ac.id

Abstract. Phishing is one of the growing cybersecurity threats, including through SMS, known as smishing. This research aims to build a model for SMS phishing detection using a machine learning approach optimized through hyperparameter tuning techniques. The data used is obtained from personal SMS messages collected through questionnaires, which information security experts then label. The SMS text is cleaned using Natural Language Processing (NLP) techniques and represented using the TF-IDF method. Ten classification algorithms are tested in this study: K-NN, Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, AdaBoost, Bagging, ExtraTree, Gradient Boosting, and XGBoost. Hyperparameter optimization is performed using Grid Search and Optuna, and performance is evaluated using accuracy, F1-score, and ROC-AUC metrics. The results show that the SVM and Logistic Regression models performed the best, achieving accuracy up to 98.5%. Hyperparameter optimization techniques have proven effective in improving the performance of SMS phishing classification models. This research is expected to contribute to the development of accurate and efficient SMS phishing detection systems.

Keywords: *Phishing*, SMS, *Hyperparameter Optimization*, *Machine Learning*, TF-IDF, *Grid Search*, Optuna

Received January 2025 / **Revised** May 2025 / **Accepted** June 2025

INTRODUCTION

Phishing is a serious threat at present [1]. The high rate of phishing attacks remains constant every year across various organizational fields [1]. Researchers have employed various methods to prevent phishing attacks by optimizing techniques to enhance detection. Optimization methods include utilizing the Earthworm Optimization Algorithm (EWA) for detecting phishing emails [2], using the African Vulture Optimization Algorithm (AVOA) for phishing website detection [3], using Hunger Search Optimization for website detection [4], applying hyperparameter optimization for website detection [5], and utilizing the Particle Swarm Optimization technique (PSO) for phishing URL detection [6], [7].

According to [1], SMS phishing is the most common attack encountered by organizations despite the implementation of information security awareness campaigns. Therefore, researchers developed an SMS phishing detection model using hyperparameter optimization techniques in machine learning. However, this optimization technique is rarely found in research on SMS phishing detection. Optimization techniques are only found in research [5]. [5] uses hyperparameter optimization techniques for phishing website detection, namely Grid Search and Genetic Algorithm. [5] uses Long Short-Term Memory (LSTM), Deep Neural Network (DNN), and convolutional neural network (CNN) techniques to implement optimization. The best accuracy is 97.37%, with an accuracy improvement of 0.1% to 1%. In contrast, [4] used CNN and LSTM techniques that were optimized using HSO. The optimization performed by [4] successfully achieved a performance score of 98.07% in accuracy, precision, recall, F1-score, and AUC. [3] used AVOA to optimize the best features for phishing website detection using LSTM and CNN. The accuracy achieved on the UCI, TAN, and PhishTank datasets was 98.87%, 99.37%, and 98.79%, respectively. [6] utilized the PSO technique to optimize the parameters used to minimize calculation errors in the artificial neural network system. The classification techniques used are Naïve Bayes, Multi-layer Perceptron, J48 Tree, LMT, Random Forest, Random Tree, C4.5, ID3, C-RT, and K-Nearest Neighbor. Almost all classification techniques yield results above 91%, unlike [7], which uses PSO as a weighting method based on the importance of feature contributions in identifying phishing from legitimate websites. The classification techniques used are back-propagation neural network (BPNN), support vector machine (SVM), k-Nearest

DOI: 10.24014/coreit.v11i1.35547

neighbor (kNN), decision tree (C4.5), random forest (RF), and naïve Bayes classifier (NB). The use of PSO can improve the accuracy of phishing classification; however, it requires a considerable amount of time to perform the weighting process [7]. [2] uses the EWA optimization technique on Naive Bayes (NB), Deep Belief Network, and Neural Network (NN). The optimization performed by [2] achieves an accuracy of 0.8571, sensitivity of 0.8182, and specificity of 0.88.

The optimizations performed by these researchers have successfully improved phishing detection techniques. However, the optimizations performed by these researchers still have weaknesses that can be improved, such as the lack of actual and up-to-date data for phishing emails [2], [7], the need for comparison with other classification algorithms [6], and the absence of hyperparameter optimization [3], [5]. Based on the recommendations [8] regarding optimization in phishing research and the weaknesses of the research presented in the background above, the researcher proposes a phishing SMS detection model using hyperparameter techniques in machine learning.

Therefore, this study proposes a phishing SMS detection model that addresses the issues encountered by the researchers, specifically by implementing hyperparameter optimization techniques. The research data used in this study comes from individuals' SMS messages that are considered phishing or not. This research model will employ 10 classification techniques: K-NN, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, AdaBoost, Bagging, Extra Trees, Gradient Boosting, and XGB. The model will be evaluated using accuracy, F1-score, and ROC-AUC to obtain the best hyperparameters.

METHODS

In recent years, machine learning techniques have been widely used to solve various problems, particularly for identifying phishing attacks [9]. Several pieces of information are required to train these machine learning techniques, including page content, URLs, and networks. Some machine learning techniques widely used by phishing researchers include KNN, Logistic regression, Support Vector Machines, Decision Trees, Random Forests, AdaBoost, Bagging, Extra Trees, Gradient Boosting, and XGB.

KNN classification is a supervised learning method based on examples that works well with distance-sensitive data [10]. K-NN can be used to detect phishing URLs [10], [11]. Logistic Regression is also used to detect phishing websites [12] and phishing URLs [13]. Logistic regression helps identify the relationship between a categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function. Support Vector Machine can be used to detect phishing URLs [14], phishing SMS and emails [15], phishing emails [16], and phishing websites [17]. One of the advantages of SVM is its ability to achieve higher accuracy with fewer samples [15]. Phishing email detection can also be performed using the Decision Tree technique [18]. Additionally, the Random Forest technique is widely used for detecting phishing SMS [19], phishing URLs [20], and phishing websites [21], [22]. Random Forest is an ensemble learning technique that creates multiple decision trees, where each tree contributes one vote to the most frequent class assignment based on the input data [22]. The AdaBoost (Adaptive Boosting) technique can be used for detecting phishing websites [23], [24]. The purpose of Adaptive Boosting is to increase the weight and selection probability of complex samples that are closer to the classification boundary, making it more challenging for the classifier to handle them [24]. The Bagging technique can be used for detecting phishing websites [25], [26]. Additionally, the ExtraTree technique can be used to detect phishing URLs [27] and phishing websites [28]. Similarly, the Gradient Boosting technique can be used to detect phishing websites [29], and the XGB technique is used by [30]. Furthermore, the application of Natural Language Processing (NLP) techniques can aid in the creation of new features, particularly for phishing emails or phishing SMS messages [31].

The methodology used to research the SMS phishing detection model with Hyperparameter Optimization in machine learning is illustrated in Figure 1.

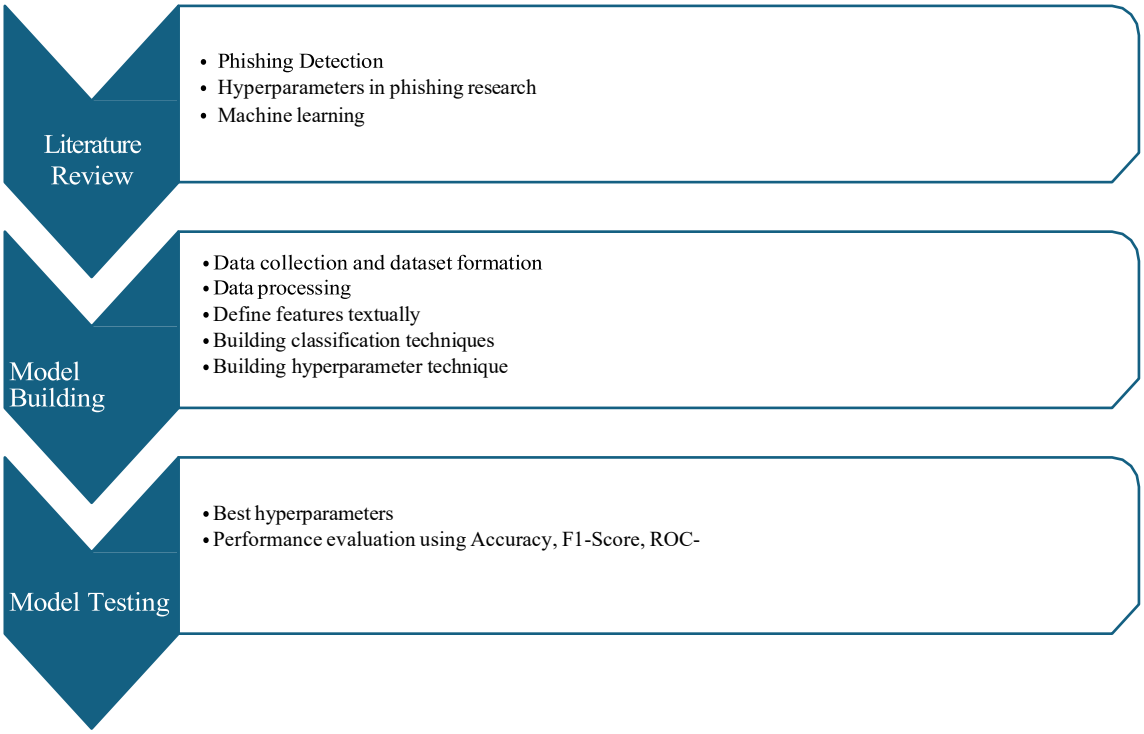

Figure 1. Research methodology used

Literature Review

We conducted a literature review of articles indexed in Scopus and Web of Science, using the following criteria: phishing detection, hyperparameters, and machine learning. The literature review aimed to identify research problems, objectives, and the latest related studies. We selected references from the last five years to ensure the inclusion of current research and to provide a significant contribution to the field of phishing detection research.

Model Building

After conducting a literature review, the researchers developed a proposed model, as shown in Figure 2, to detect phishing SMS using optimization techniques in machine learning.

The dataset used in this research was collected from personally owned SMS. Researchers created a questionnaire containing examples of SMS suspected of phishing and then distributed it to users to solicit sample SMS similar to those in the questionnaire. After the SMS suspected of phishing is obtained, the next step is to label which SMS are phishing and which are not by information security experts. Then, the labeling results will undergo a text cleaning process using NLP techniques, such as removing hyperlinks and markup, numbers, emojis, spaces, symbols, and punctuation marks, as well as applying stopword and stemming techniques. After text cleaning, the next step is to assign weights to the text using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. TF-IDF will determine which texts have a relationship with each other, and which is the strongest in a corpus. The results of TF-IDF will be processed by 10 classification techniques, namely K-NN, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, AdaBoost, Bagging, ExtraTree, Gradient Boosting, and XGB. Classification techniques are parameterized to get the best parameters using hyperparameter techniques.

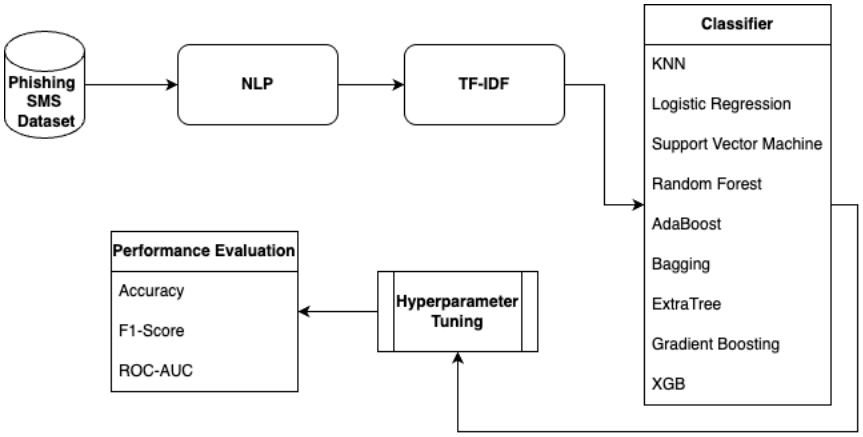

Figure 2. Proposed SMS phishing detection model with hyperparameter technique on machine learning

Model Testing

The proposed model will be evaluated using performance measures, including accuracy, F1-score, and ROC-AUC, to determine the optimal output and hyperparameters for each classification technique.

RESULT AND DISCUSSION

Comparison of 1500 Data Classification Models with Grid Search CV

Table 1. Performance Evaluation of 10 Classification Models with 1500 Data

Model	Accuracy	F1-Score	ROC-AUC
K Neighbors Classifier	0.9775	0.9723	0.9736
Random Forest	0.9625	0.9526	0.9417
Logistic Regression	0.9725	0.9656	0.9594
SVC	0.9850	0.9815	0.9815
Decision Tree	0.9725	0.9669	0.9782
There is a Boost Classifier	0.9775	0.9725	0.9763
Bagging Classifier	0.9700	0.9638	0.9737
Extra Trees Classifier	0.9475	0.9320	0.9124
XGB Classifier	0.9775	0.9728	0.9816

Based on Table 1, the performance evaluation results of the 10 classification models tested indicate that the Support Vector Classifier (SVC) displays the best performance, with an accuracy of 98.50%, an F1-score of 98.15%, and a ROC-AUC of 98.15%. This indicates that both models can consistently predict positive and negative classes very well. In addition, the XGB Classifier model also yields competitive results, with an accuracy of 97.75%, an F1-score of 97.28%, and the highest ROC-AUC value among all models, at 98.16%. This value shows XGBoost's ability to accurately distinguish between the two target classes, making this model a strong alternative to consider.

Meanwhile, the K-Nearest Neighbors Classifier model also performs very well, with an accuracy of 97.75%. However, this model has an inherent weakness as a non-parametric algorithm that may be less efficient on larger datasets. On the other hand, Random Forest demonstrates good performance with an accuracy of 96.25%. However, the recall value for class 0 is slightly lower than that for other classes, indicating that this model has some difficulty in predicting minority data. The Decision Tree and Bagging

Classifier models also exhibit relatively stable performance, with an accuracy of around 97%. However, they still lag in terms of precision and F1-score compared to models such as SVC or XGBoost.

The models with lower performance are the Extra Trees Classifier and the Gradient Boosting Classifier. Although both models provide an accuracy of over 94%, they have weaknesses in correctly predicting class 0. This is evident from the lower precision and recall values for that class, resulting in both F1-score and ROC-AUC being lower than those of other models. The ExtraTreesClassifier, in particular, has an accuracy of 94.75% and an ROC-AUC of 91.24%, indicating that this model still has room for improvement, primarily through further parameter tuning. Overall, the SVC model is the top choice as it achieves the highest performance across all evaluation metrics.

Comparison of 500 Balanced Data Classification Models with Grid Search CV

Based on the evaluation results in Table 2, Logistic Regression and SVC are the two models with the best performance, recording the highest accuracy, F1-Score, and ROC-AUC (0.985). Both models are suitable for data with clear class margins, with Logistic Regression being superior in terms of simplicity and speed. On the other hand, ensemble models such as Random Forest, AdaBoost, and Gradient Boosting yield nearly equivalent results (accuracy 0.975–0.98) and excel at handling complex data or data with non-linear patterns. However, the complexity and training time of ensemble models are higher than those of Logistic Regression. Meanwhile, Decision Tree demonstrates high performance (with an accuracy of 0.98), but is more prone to overfitting. Overall, Logistic Regression is the ideal choice for problems prioritizing efficiency and interpretability, while SVC or ensemble models are more suitable for data with more complex patterns.

Table 2. Performance Evaluation of 10 Classification Models with 500 Balanced Data

Model	Accuracy	F1-Score	ROC-AUC
K Neighbors Classifier	0.9700	0.9699	0.9692
Random Forest	0.9750	0.9749	0.9740
Logistic Regression	**0.9850**	**0.9850**	**0.9844**
SVC	**0.9850**	**0.9850**	**0.9848**
Decision Tree	0.9800	0.9800	0.9800
There is a Boost Classifier	0.9800	0.9799	0.9796
Bagging Classifier	0.9800	0.9799	0.9796
Extra Trees Classifier	0.9750	0.9749	0.9740
Gradient Boosting	0.9800	0.9799	0.9792
XGB Classifier	0.9750	0.9749	0.9740

Evaluation of KNN Classification of 500 Balanced Data with Optuna

Based on Table 3, the evaluation results of the KNeighborsClassifier model optimized using Optuna indicate that the optimal hyperparameter is n_neighbors = 4. We use Optuna because it saves time and money, especially with its excellent auto-pruning feature. Optuna is a specialized framework for automating hyperparameter tuning. This model demonstrates excellent performance with an accuracy of 97.5%, an F1-score of 97.49%, and an ROC-AUC of 97.44%, reflecting its high ability to distinguish between positive and negative classes.

Table 3. Evaluation of Classification Results of 500 Balanced Data with Optuna

Metric	Value
Accuracy	0.975
F1-Score	0.9749
ROC-AUC	0.9744
Best Trial	2
Best Value (1 - F1)	0.0188

In detail, for the positive class (1), the model has a precision of 0.99 and a recall of 0.96, while for the negative class (0), the precision is 0.96 and the recall is 0.99. This indicates that the model is highly accurate

in predicting the positive class, although some positive data points are still not detected. Conversely, for the negative class, almost all data points are classified correctly. Optuna successfully identified the optimal value of the k parameter using the F1-Score as an evaluation benchmark, with the best value obtained in trial 2, yielding a minimum (1 - F1) value of 0.0188. Overall, this model exhibits balanced and accurate performance in classification tasks, characterized by a low error rate and an optimal balance between precision and recall.

Comparison of KNN Classification Evaluation with Grid Search Optimization and Optuna

Table 4. Comparison of KNN with Grid Search and Optuna

KNN Model	Accuracy	F1-Score	ROC-AUC
Grid Search	0.9700	0.9699	0.9692
Optuna	0.975	0.9749	0.9744

Based on Table 4, which compares the performance of the KNN model between Grid Search and Optuna, it can be concluded that Optuna shows better results in all evaluation metrics (Accuracy, F1-Score, and ROC-AUC). The accuracy on Optuna is recorded at 0.975, slightly higher than Grid Search, which reaches 0.9700. This suggests that Optuna can identify a more optimal combination of parameters, leading to more accurate predictions.

Additionally, the F1-score on Optuna reached 0.9749, while Grid Search only achieved 0.9699. This improvement indicates that Optuna provides a better balance between precision and sensitivity, especially on data with potentially imbalanced class distributions. This capability is crucial to ensure that the model is not only accurate overall but also effective in classifying minority classes.

In terms of ROC-AUC, Optuna also outperforms Grid Search with a score of 0.9744 compared to 0.9692. A higher ROC-AUC value indicates that the model tuned using Optuna is more reliable in distinguishing between positive and negative classes, which is an important indicator of model stability and reliability.

Overall, Optuna proved to be superior to Grid Search in the parameter tuning process. This is likely due to the use of adaptive search algorithms such as Tree-structured Parzen Estimators (TPE), which are more efficient in exploring the parameter space than traditional grid searches. Therefore, Optuna is recommended as a more effective parameter tuning method for KNN models, especially when model performance is a top priority.

Comparison of KNN Classification Evaluation with PSO, Grid Search and Optuna Optimization

Table 5. Comparison of KNN evaluation with PSO-based feature weighting, Grid Search, and Optuna approaches

	Accuracy (%)		
Model	PSO-based feature weighting	Grid Search	Optuna
KNN	96	97	97.5

Based on Table 5, which shows a comparison of the performance of the KNN model with PSO-based feature weighting, Grid Search, and Optuna approaches, the following is an analysis and discussion in paragraph form:

The application of PSO-based feature weighting achieved an accuracy of 96% for the KNN model, which is lower than parameter tuning methods such as Grid Search (97%) and Optuna (97.5%). These results suggest that utilizing PSO (Particle Swarm Optimization) for feature weighting can enhance model performance. However, it still lags behind parameter tuning based on Grid Search and Optuna. The performance of Grid Search and Optuna is superior due to their ability to find the best parameter combinations for the KNN model. Optuna, with an accuracy of 97.5%, yields the best results among the three methods. This is due to more efficient adaptive parameter search algorithms such as Tree-structured Parzen Estimators (TPE), which allow for more optimal exploration of the parameter space compared to Grid Search.

Overall, the Optuna approach is the most effective for improving the accuracy of the KNN model. At the same time, the PSO-based feature weighting still makes a significant contribution, having a positive effect on model quality. Thus, a combination of methods such as PSO for feature optimization and Optuna for parameter tuning has the potential to provide better results when applied together.

CONCLUSION

Based on the analysis results, the SVC (Support Vector Classifier) model exhibits the best performance for data sizes of 1500 and 500, achieving the highest accuracy, F1-Score, and ROC-AUC (98.50%). This model excels in consistently predicting positive and negative classes, making it the top choice for data with clear class margins. Additionally, Logistic Regression also delivers excellent results, with comparable performance on datasets of 500 instances and advantages in simplicity and efficiency. Ensemble models, such as Random Forest, AdaBoost, and Gradient Boosting, show competitive results, particularly for complex datasets or those with non-linear patterns, although their training times are longer.

Meanwhile, the KNeighborsClassifier (KNN) optimized using Optuna shows improved performance compared to Grid Search, with an accuracy of 97.5% versus 97%. Optuna proves to be more effective in finding optimal parameter combinations. The PSO-based Feature Weighting approach contributes to model quality but still lags behind parameter tuning methods, such as Optuna. Combining PSO for feature optimization and Optuna for parameter tuning has the potential to produce more accurate models. The obstacle we encountered during this research process was that we were unable to process the entire algorithm that we presented at the beginning, because we wanted to focus on algorithms with low detection techniques and then optimize them to improve their performance. Overall, SVC is the best choice for achieving the highest performance, while Logistic Regression and Optuna are efficient options for applications that require speed and accuracy.

REFERENCES

[1] Proofpoint, “2023 State of the Phish,” 2023. [Online]. Available: https://www.proofpoint.com/us/resources/threat-reports/state-of-phish

[2] M. Arshey and K. S. Angel Viji, “An optimization-based deep belief network for the detection of phishing e-mails,” *Data Technologies and Applications*, vol. 54, no. 4, pp. 529–549, Aug. 2020, doi: 10.1108/DTA-02-2020-0043.

[3] M. A. Elberri, Ü. Tokeşer, J. Rahebi, and J. M. Lopez-Guede, “A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA),” *Int J Inf Secur*, vol. 23, no. 4, pp. 2583–2606, Aug. 2024, doi: 10.1007/s10207-024-00851-x.

[4] H. Shaiba, J. S. Alzahrani, M. M. Eltahir, R. Marzouk, H. Mohsen, and M. A. Hamza, “Hunger Search Optimization with Hybrid Deep Learning Enabled Phishing Detection and Classification Model,” *Computers, Materials and Continua*, vol. 73, no. 3, pp. 6425–6441, 2022, doi: 10.32604/cmc.2022.031625.

[5] M. Almousa, T. Zhang, A. Sarrafzadeh, and M. Anwar, “ Phishing website detection: How effective are deep learning-based models and hyperparameter optimization? ,” *SECURITY AND PRIVACY*, vol. 5, no. 6, Nov. 2022, doi: 10.1002/spy2.256.

[6] S. M. Alshahrani, N. A. Khan, J. Almalki, and W. Al Shehri, “URL Phishing Detection Using Particle Swarm Optimization and Data Mining,” *Computers, Materials and Continua*, vol. 73, no. 3, pp. 5625–5640, 2022, doi: 10.32604/cmc.2022.030982.

[7] W. Ali and S. Malebary, “Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection,” *IEEE Access*, vol. 8, pp. 116766–116780, 2020, doi: 10.1109/ACCESS.2020.3003569.

[8] R. Abdillah, Z. Shukur, M. Mohd, T. S. M. Z. Murah, I. Oh, and K. Yim, “Performance Evaluation of Phishing Classification Techniques on Various Data Sources and Schemes,” *IEEE Access*, vol. 11, pp. 38721–38738, 2023, doi: 10.1109/ACCESS.2022.3225971.

[9] O. K. Sahingoz, E. BUBEr, and E. Kugu, “DEPHIDES: Deep Learning Based Phishing Detection System,” *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[10] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network," *Wirel Commun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/8241104.

[11] E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, "A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators," *Sensors*, vol. 23, no. 9, p. 4403, Apr. 2023, doi: 10.3390/s23094403.

[12] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Phishing website detection: An improved accuracy through feature selection and ensemble learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, pp. 252–257, 2019, doi: 10.14569/IJACSA.2019.0100133.

[13] G. S. and N. B. P.K., "Analysis of Phishing Detection Using Logistic Regression and Random Forest," *Journal of Applied Information Science*, vol. 8, no. 1 & 2, pp. 7–13, 2020, [Online]. Available: http://www.publishingindia.com

[14] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, Feb. 2022, doi: 10.1016/j.jksuci.2019.12.005.

[15] U. Maqsood, S. Ur Rehman, T. Ali, K. Mahmood, T. Alsaedi, and M. Kundi, "An Intelligent Framework Based on Deep Learning for SMS and e-mail Spam Detection," *Applied Computational Intelligence and Soft Computing*, vol. 2023, pp. 1–16, Sep. 2023, doi: 10.1155/2023/6648970.

[16] 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*, Jun. 2022, doi: 10.1007/s40747-022-00760-3.

[17] M. Sabahno and F. Safara, "ISHO: improved spotted hyena optimization algorithm for phishing website detection," *Multimed Tools Appl*, vol. 81, no. 24, pp. 34677–34696, Oct. 2022, doi: 10.1007/s11042-021-10678-6.

[18] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Comput*, vol. 25, no. 6, pp. 3819–3828, Dec. 2022, doi: 10.1007/s10586-022-03604-4.

[19] S. R. A. Samad, P. Ganesan, J. Rajasekaran, M. Radhakrishnan, H. Ammaippan, and V. Ramamurthy, "SmishGuard: Leveraging Machine Learning and Natural Language Processing for Smishing Detection," *IJACSA*, vol. 14, no. 11, 2023, doi: 10.14569/IJACSA.2023.0141160.

[20] A. K. Jain, N. Debnath, and A. K. Jain, "APuML: An Efficient Approach to Detect Mobile Phishing Webpages using Machine Learning," *Wirel Pers Commun*, 2022, doi: 10.1007/s11277-022-09707-w.

[21] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," *Sensors*, vol. 21, no. 24, p. 8281, Dec. 2021, doi: 10.3390/s21248281.

[22] S. Wang, S. Khan, C. Xu, S. Nazir, and A. Hafeez, "Deep Learning-Based Efficient Model Development for Phishing Detection Using Random Forest and BLSTM Classifiers," *Complexity*, vol. 2020, pp. 1–7, Sep. 2020, doi: 10.1155/2020/8694796.

[23] Z. G. Al-Mekhlafi *et al.*, "Phishing websites detection by using optimized stacking ensemble model," *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 109–125, 2022, doi: 10.32604/csse.2022.020414.

[24] O. Rahmani Seryasat *et al.*, "Recognizing phishing websites based on a bayesian combiner," *Int. J. Nonlinear Anal. Appl*, vol. 12, pp. 2008–6822, 2021, doi: 10.22075/IJNAA.2021.5457.

[25] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites," *IEEE Access*, vol. 8, pp. 142532–142542, 2020.

[26] M. Al-Sarem *et al.*, "An Optimized Stacking Ensemble Model for Phishing Websites Detection," *Electronics (Basel)*, vol. 10, no. 11, p. 1285, May 2021, doi: 10.3390/electronics10111285.

[27] H. Bouijij, A. Berqia, and H. Saliah-Hassan, "Phishing URL classification using Extra-Tree and DNN," in *2022 10th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, Jun. 2022, pp. 1–6. doi: 10.1109/ISDFS55398.2022.9800795.

[28] M. K. Pandey, M. K. Singh, S. Pal, and B. B. Tiwari, "Prediction of phishing websites using machine learning," *Spatial Information Research*, vol. 31, no. 2, pp. 157–166, Apr. 2023, doi: 10.1007/s41324-022-00489-8.

[29] R. Pavan, M. Nara, S. Gopinath, and N. Patil, "Bayesian optimization and gradient boosting to detect phishing websites," in *2021 55th Annual Conference on Information Sciences and Systems, CISS 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021. doi: 10.1109/CISS50987.2021.9400317.

[30] A. Aljofey *et al.*, "An effective detection approach for phishing websites using URL and HTML features," *Sci Rep*, vol. 12, no. 1, p. 8842, Dec. 2022, doi: 10.1038/s41598-022-10841-5.

[31] P. Bountakas, K. Koutroumpouchos, and C. Xenakis, "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, Vienna Austria: ACM, Aug. 2021, pp. 1–12. doi: 10.1145/3465481.3469205.