Improving Web Security through Machine Learning: A Feature-Based Methodology for Detecting Phishing URLs

Reem Alzubi
Engineering and Artificial Intelligence Department, Al-Salt Technical College, Al-Balqa Applied University, Al-Salt, Jordan
reem.alzoubi@bau.edu.jo

Tariq Bishtawi
Department of Computer Science, Amman Arab University, Amman 11953, Jordan
t.bishtawi@aau.edu.jo (corresponding author)

Hassan Kassem
Department of Communications and Computer Networks, Arab University College of Technology, Amman, Jordan
hassanluther@gmail.com

Received: 8 May 2025 | Revised: 28 June 2025 and 6 July 2025 | Accepted: 8 July 2025

ABSTRACT

Phishing attacks remain a significant and evolving threat to web security, often using malicious URLs to deceive users into sharing personal information. This study employs a detailed, feature-based approach to develop a machine learning method for detecting phishing URLs. The analysis includes four advanced machine learning classifiers that utilize comprehensive features from lexical patterns, host-based, and content-based URL characteristics. These classifiers are Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, and Extreme Gradient Boosting (XGBoost). Results demonstrate that ensemble methods outperform individual models in phishing detection, with XGB and RF achieving higher accuracy, precision, and recall across all metrics. These findings contribute to the development of real-time phishing detection tools, although effective feature engineering and model selection remain crucial for enhancing internet security.

Keywords-phishing URLs; RF; DT; SVM;extreme gradient boosting; phishing detection

I. INTRODUCTION

Phishing remains one of the most significant cyber threats, largely due to the rapid growth of digital communication and online commerce. Phishing schemes trick users into visiting fake websites designed to steal sensitive information, such as usernames, passwords, and financial details. The techniques employed in phishing attacks continually evolve at a rate that surpasses the capabilities of conventional security measures such as blacklists and heuristic filters. The current situation demands intelligent, adaptable security systems because of clear upcoming needs. The cybersecurity sector now predominantly relies on Machine Learning (ML) as its primary differentiator in both detecting and defending against phishing attacks. By analyzing patterns and historical data, ML models can identify potentially malicious URLs before traditional detection systems can detect them. This research aims to contribute to this expanding field by developing and evaluating a feature-based plan for identifying phishing URLs utilizing four well-known ML techniques: RF, DT, SVM (with an RBF kernel), and XGBoost.

ML plays a vital role in combating phishing, with innovation serving as the key driver to enhance online security. The study by [1] examines the effectiveness of ML techniques in detecting phishing websites. Researchers used randomly labeled datasets containing genuine and fraudulent URLs to train models, including Logistic Regression (LR), DT (principal partial derivative of Monte Carlo), RF, AdaBoost, and XGBoost. The RF model was the best performer, achieving an impressive 89.8 percent success rate. The research confirms that combining ensemble learning methods with feature optimization significantly improves phishing detection systems. Using this analogy, researchers gained essential concepts that could help enhance security regulations in cyberspace.

In [2], the authors improved phishing detection by combining URL and hyperlink features into a single hybrid feature set. The researchers chose RF and XGBoost classifiers to analyze only client-side data. This approach differs from traditional methods that depend on third-party validation through search engines, as it achieves higher zero-hour phishing attack detection rates. The combination of hyperlink behavior and URL structural indicators yields a model that achieves 96.81% accuracy in identification. The solution offers real-time phishing detection, providing practical advantages in situations where external verification resources are unavailable. Researchers in [3] developed a phishing detection system that uses ML to enhance classification accuracy by combining HTML elements, domain features, and URL characteristics. This research integrates three ensemble models (RF, SVM, and XGBoost) using stacking techniques to create a more reliable system that reduces false positives. Web features extracted from the system help distinguish between legitimate websites and malicious ones by analyzing domain age, SSL validity, and keywords associated with phishing. The findings show strong potential for real-time deployment, with the model achieving an accuracy of 86.2% and a precision rate of 92% in detecting phishing websites. The system supports practical cybersecurity applications through a Flask-based interface and a community reporting system, enabling scalable deployment.

Authors in [4] studied phishing detection using ML models that analyzed URL features, including domain age, URL length, and HTTPS usage. They used RF, DT, and SVM algorithms, applying stratified 10-fold cross-validation to minimize bias. RF demonstrated the highest accuracy at 97%, while SVM achieved a precision of 0.92 and a recall of 0.95, as indicated by the results. This paper demonstrates the ability of neural network models to enhance phishing detection systems and shows that ensemble methods are effective tools for detection. According to the research findings, future cybersecurity solutions will need to include explainable artificial intelligence alongside efficient feature selection processes. In [5], this research study examined the application of RF, LightGBM, and XGBoost algorithms for detecting fake websites. The methodology classifies websites by analyzing lexical features and metadata, including URL length and keyword content such as "gestive," to differentiate between benign sites, malware, phishing websites, and defacement attacks. RF achieved a 97% accuracy, surpassing both LightGBM and XGBoost. The research indicates that well-designed features enhance detection accuracy, and ensemble methods provide an effective approach to internet security. The model's real-time prediction ability makes it appealing for the cybersecurity market. Authors in [6] explore how the integration of ML algorithms with address bar features effectively detects phishing links. This study evaluates three classifiers by analyzing an evenly mixed set of phishing and legitimate URLs from PhishTank and the Canadian Institute of Cybersecurity. The models evaluated encompass SVM, RF, and DT. The assessment of URLs is based on eight fundamental features, including URL length, IP address detection, the utilization of URL shortening services, and the identification of special characters. The SVM model demonstrated the highest performance, with the DT and RF models attaining an accuracy of 84.1% in detection results. According to this study, the effectiveness of phishing detection critically depends on feature ranking as a vital element. The research in [7] presents a new 1D Convolutional Neural Network (CNN) system for phishing detection URLs. The system is designed to create advanced methods for identifying malicious URLs, enhancing protection of computer networks from phishing threats. This model attains an outstanding success rate of 99.85%, surpassing all existing methodologies in the detection of phishing URLs. These high accuracy levels demonstrate the model's ability to effectively distinguish between genuine URLs and phishing domains. The study's ability to explain its results is a key contribution to the field. SHapley Additive exPlanations (SHAP) is used by the authors to interpret the model’s predictions by quantifying the contribution of each input feature to the final output. This enables users to identify which features have the most significant impact on detecting phishing URLs. Authors in [8] created a new system for identifying phishing URLs by combining BERT feature extraction with Deep Learning (DL). This technique enhances standard methods for phishing detection by increasing accuracy rates. The research employs Natural Language Processing (NLP) to analyze URLs by extracting relevant features for improved interpretation and contextual understanding. The capability to distinguish genuine web pages from phishing attempts relies on this feature. The implementation model attained an accuracy of 96.66% in identifying phishing URLs. The outstanding accuracy demonstrates that NLP and DL systems work well together in safeguarding cybersecurity. The research report states that post-processing decreased the dataset to 472,259 records. However, the paper does not specify any boundaries that might restrict dataset availability.

Authors in [9] identified three key features, "length_url," "time_domain_activation," and "Page_rank," that determine how well phishing attempts can be detected. Feature selection emerges as a crucial factor in detecting these threats, according to the study's findings. This research evaluates three ML approaches: CatBoost, XGBoost and Explainable Boosting Machine (EBM) to assess their effectiveness in accurately detecting phishing activity. SHAP is used to provide visual and interpretable explanations of the model’s predictions, helping stakeholders understand the reasoning behind phishing detection and increasing trust in automated systems. The study currently analyzes URL-based features, but incorporating WHOIS information and website content analysis is expected to significantly enhance the accuracy of phishing detection. According to [10], a ML model was developed that accurately distinguished legitimate websites from phishing targets with 96.9% accuracy. The model's features demonstrate significant potential for application in real-world scenarios. The researchers improved detection accuracy by combining analysis of URL structures, domain attributes, and website content during development. They emphasize that evaluating the model's performance needs multiple metrics, as relying only on accuracy is insufficient. They also highlight the importance of measuring precision and recall. The paper does not demonstrate how the model operates in real-time scenarios, which is crucial for rapid phishing detection and prevention.

The primary objective of this study is to develop a robust and efficient ML-based system for detecting phishing URLs, thereby enhancing web security. The research aims to identify and extract a comprehensive set of URL-based features that effectively differentiate between legitimate and phishing websites. Using these features, several ML models are developed and trained, including RF, DT, SVM with a RBF kernel, and XGBoost. The models are evaluated using standard performance metrics, including accuracy, precision, recall, F1-score, and computational efficiency. This study addresses key research questions, including which URL characteristics are most effective for distinguishing malicious from legitimate sites, how different classifiers compare in terms of predictive performance, whether ensemble methods can outperform individual models in phishing detection, and what impact these models may have on latency and computational efficiency when deployed in real-time detection systems.

II. METHODOLOGY

In our proposed approach for detecting phishing URLs, we employ a set of advanced ML classifiers, including RF, DT, SVM, and XGBoost. This methodology is designed to enhance web security by systematically processing and analyzing URL data to distinguish between legitimate and malicious websites. The following is a concise overview of the key steps involved in the proposed detection framework:

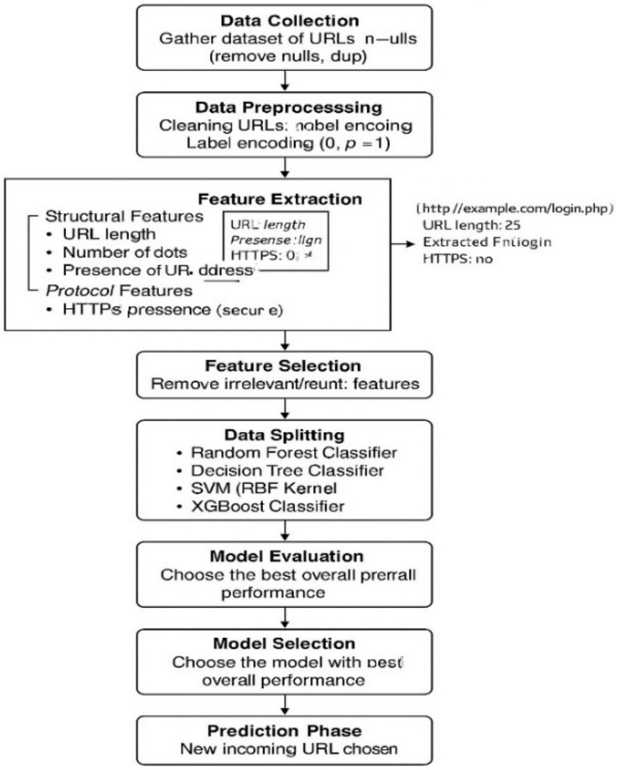

Fig. 1. The proposed model.

A. Data Collection

In recent years, phishing attacks have grown increasingly sophisticated, making them more difficult to detect. According to a recent survey conducted by Intel, an alarming 97% of security professionals reported difficulty in distinguishing between legitimate and phishing URLs [11]. The dataset in [12] contains 11,430 URLs, each characterized by 87 extracted features. This dataset serves as a benchmark for ML-based phishing detection systems. Its features are divided into three categories: seven are obtained through querying other services, twenty-four derive from the content of relevant pages, and fifty-six are based on the structure and syntax of URLs. Interestingly, the dataset is evenly balanced, with exactly 50% of the URLs being phishing attempts and 50% being legitimate [13].

B. Data Preprocessing

To prepare the "Phishing URLs Dataset" for use in ML models, it is advisable to implement several preprocessing techniques. Begin by removing any URLs that are improperly formatted. Then, extract useful details from the URLs, such as their length, the number of special characters, or the presence of specific keywords [14]. To ensure consistency, it's important to normalize your data. The first step is to remove extra space and convert all URLs to lowercase. To enable ML algorithms to comprehend the data, it is necessary to convert categorical values into numerical labels, such as "legitimate" and "phishing." A dataset is considered imbalanced when one class contains significantly more instances than the others, which can lead to biased model performance. In such cases, it may be necessary to apply data balancing techniques to address this issue [15]. The preprocessing stage optimizes the dataset for ML, enabling effective identification of phishing URLs. The selection of features and modeling techniques should be aligned with the specific characteristics of the dataset.

C. Machine Learning Model Training

The ensemble method, which combines RF, SVM, DT, and XGBoost, effectively classifies URLs. Autoencoder Neural Networks utilize their capacity to learn website distribution patterns to detect abnormal features in phishing sites.

1) Random Forest

The RF Classifier is a key method for identifying phishing URLs. The model uses URL details to differentiate between phishing sites and legitimate ones. The RF classifies URLs by combining multiple decision trees to generate its results. It performs better with large datasets than individual decision trees and shows stronger performance when fitting data. The ability of RF to recognize non-linear relationships among features provides excellent results in detecting phishing attempts [16].

2) Support Vector Machines

An SVM is an effective tool for detecting phishing URLs. The method utilizes mathematical structures to distinguish between phishing and legitimate URLs, extracting clear URL features and metadata properties. SVM serve as supervised classification models, mainly used for identifying different classes. An SVM finds the most effective separating hyperplane that distinguishes phishing from safe URLs with the widest margin. With kernel methods, SVMs perform well in high-dimensional spaces and handle nonlinear data efficiently [17].

3) Decision Tree

The DT classifier offers a straightforward yet effective approach for identifying phishing URLs. It functions autonomously by learning from various URL features to distinguish between legitimate and malicious websites. As a supervised learning algorithm, the DT constructs a tree-like structure in which data is recursively split into branches based on the values of specific features [18]. At each node, the algorithm selects the optimal feature and corresponding threshold by evaluating metrics such as Gini impurity and information gain (also known as entropy), with the goal of maximizing the informativeness of each partition. This branching process continues until predefined stopping conditions are met, such as reaching the maximum tree depth or the minimum number of samples required at a leaf node [19].

4) XGBoost

ML often uses XGBoost for phishing URL detection because it is a highly effective implementation of extreme gradient boosting. This gradient boosting ensemble method has become popular in cybersecurity for its ability to handle structured data, tabular datasets, and URL-derived feature vectors. XGBoost constructs layered decision trees sequentially, refining the model to address previous limitations. It outperforms traditional gradient boosting in terms of speed and overfitting prevention through techniques such as parallel processing, regularization, and sparse data handling [20].

III. RESULT ANALYSIS AND DISCUSSION

This study tested four ML algorithms: RF, DT, SVM with RBF kernel, and XGBoost. The researchers extracted URL features based on structure, lexical content, and domain attributes to identify phishing conditions. All models were trained on the same dataset, and their performance was evaluated using metrics such as precision, accuracy, recall, F1 score, and training time. Our analysis confirmed the unique strengths of each method and highlighted their weaknesses in addressing the problem.

A. XGBoost: The Most Balanced Performer

XGBoost demonstrated superior performance compared to other models. It achieved a 96.6% success rate and an F1 score of 96.7%, showcasing its ability to produce balanced results. The training process was quick, lasting only 0.48 seconds, which simplifies the model's deployment. XGBoost effectively handles uneven data distribution while capturing complex prediction boundaries, with a low tendency to overfit the training data. It is especially suitable for high-stakes situations, such as phishing detection, where failure to identify, threats can lead to severe consequences. The individual performance metrics of the XGBoost model are shown in Figure 2. These metrics provide a targeted understanding of the model's behavior across key evaluation areas by illustrating the balance among accuracy, precision, recall, and F1-score.

B. Random Forest: Reliable and Interpretable

Based on the RF results, phishing URLs are detected effectively, with an accuracy of 96.3% and a recall of 97.3%, demonstrating it is a reliable model.

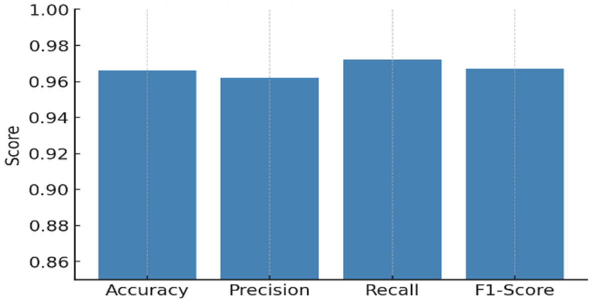

Fig. 2. Performance metrics of XGBoost

Although not quite as accurate as XGBoost, RF's interpretation is able to provide vital information in situations where that is needed. It generalizes well because it is an ensemble. However, it takes slightly longer to train than XGBoost. The individual performance measures of the RF model are shown in Figure 3. It provides a comprehensive understanding of how the model performs across key assessment criteria through the balance between accuracy, F1-score, recall, and F1-score.

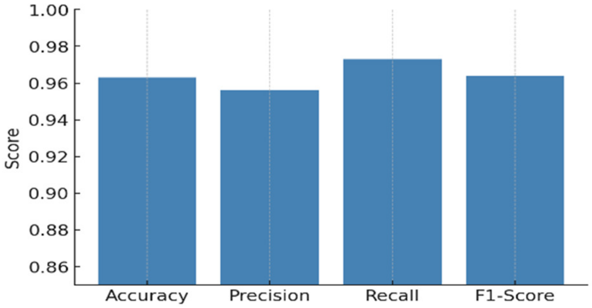

Fig. 3. Performance metrics of RF

C. Support Vector Machine: High Accuracy but Slow Training

The SVM using RBF kernel achieved an impressive F1-score of 95.3% and a very high precision score of 95.1%, indicating a low false-positive rate. Despite being the best among the four models, it had a relatively long fit time of 2.3 seconds. The SVM provided reliable performance, suggesting it could be useful in scenarios where the cost of a false positive is very high, such as labeling a genuine URL as phishing. However, due to its slower speed and lower scalability, the SVM is unlikely to be suitable for production challenges. The individual performance metrics of the SVM (RBF) model are shown in Figure 4. Its accuracy, precision, recall, and F1-score are well balanced, offering a clear understanding of how the model performs across key evaluation metrics.

D. Decision Tree: Quickest but Less Accurate

With a training time of 07 seconds, the DT classifier is well-suited for rapid prototyping or environments with limited resources. Although this method resulted in a lower F1-score of 92.5% and an accuracy of 92.3% compared to ensemble and kernel-based approaches, it appears that the propensity of standalone trees to overfit is a primary reason for its weaker performance on the test set. Figure 5 illustrates the model's behavior by displaying the DT's individual performance metrics, providing a clear perspective on how it balances accuracy, precision, recall, and F1-score across key evaluation aspects

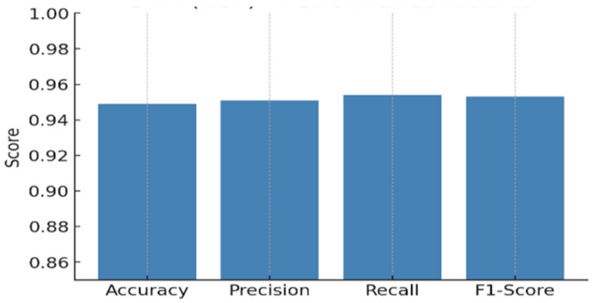

Fig. 4. Performance metrics of SVM

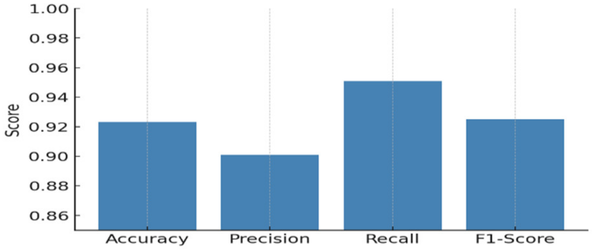

Fig. 5. Performance metrics of DT

E. Comparative Results

This section provides a comparative analysis of four ML models: RF, DT, SVM, and XGBoost, for phishing URL detection. The evaluation utilized key performance metrics, including accuracy, precision, recall, F1-score, and training time. Results are summarized in Table I, along with Figures 6 and 7, which highlight the performance differences [21]. The detailed comparison shows that XGBoost achieved the highest F1 Score and accuracy, with SVM close behind. RF ranked third overall, while DT, despite having the shortest training time, showed lower predictive performance.

TABLE I. COMPARISON OF ML MODELS FOR THE IDENTIFICATION OF PHISHING URLS

	Accuracy	**Precision**	**Recall**	**F1-score**	**Training time(s)**
Random forest	0.963	0.956	0.973	0.964	0.6
Decision tree	0.923	0.901	0.951	0.925	0.07
SVM(RBF)	0.949	0.951	0.954	0.953	2.3
XGBoost	0.966	0.962	0.972	0.967	0.48

All models were trained and evaluated under consistent conditions using scikit-learn and XGBoost libraries to ensure fairness and reproducibility. The following list details the hyperparameter settings and operational aspects employed in this study's training process.

1. RF Classifier:

- Library: sklearn.ensemble.RandomForestClassifier
- Key parameters: n_estimators = 100, max_depth = None, criterion = gini and bootstrap= True
- Features: All the 87 preprocessed features (lexical, content-based, and host-based) Divided into: 80/20 > train-test; balanced dataset.

2. Decision Tree Classifier:

- Library: Sklearn.tree.DecisionTreeClassifier
- Key parameters: criterion = entropy, max_depth = 15, min_samples_split = 10
- Applied the same data and preprocessing to the RF
- Fastest model basis performance.

3. Support Vector Machine (SVM) Classifier:

- Library: sklearn.svm.SVC
- Key parameters: kernel = 'rbf', C = 1.0, and gamma = 'scale' and probability = True.
- Standardization of the input data via StandardScaler. Remarkable in its high precision, however, it requires the longest training time.

4. XGBoost Classifier:

- Library: xgboost.XGBClassifier
- Key parameters: n_estimators = 100, max_depth = 6, learning_rate = 0.1, subsample = 0.8, colsample_bytree = 0.8, eval_metric = 'logloss
- Applied early stopping (10 rounds) in validation.
- It has the most balanced performance on its metrics.

Each model was evaluated using a 5-fold cross-validation scheme as well as on a held-out test set comprising 20 percent of the data. The reported performance metrics, including accuracy, precision, recall, F1-score, and training time, represent the average values across the cross-validation folds.

Figure 6 presents the accuracy, precision, recall, and F1-score for each model. While DT exhibits slightly lower performance, both XGBoost and RF consistently demonstrate strong results across all evaluation metrics. SVM also performs well; however, it is somewhat less efficient in terms of training time. Figure 7 illustrates the training time required for each model. DT is the fastest to train, making it well-suited for rapid prototyping. In contrast, SVM has the longest training time, which may limit its scalability in real-time applications [22].

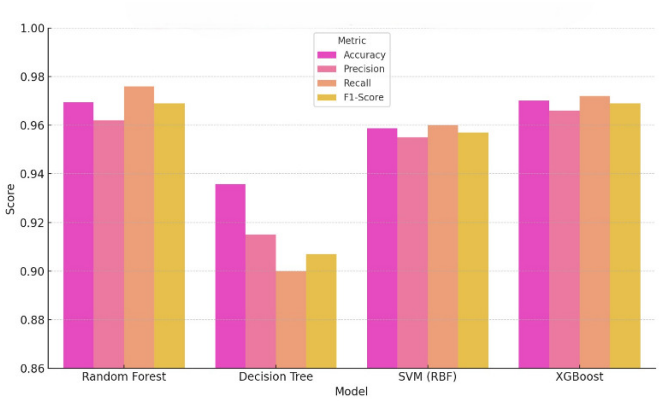

Fig. 6. Comparison of performance metrics.

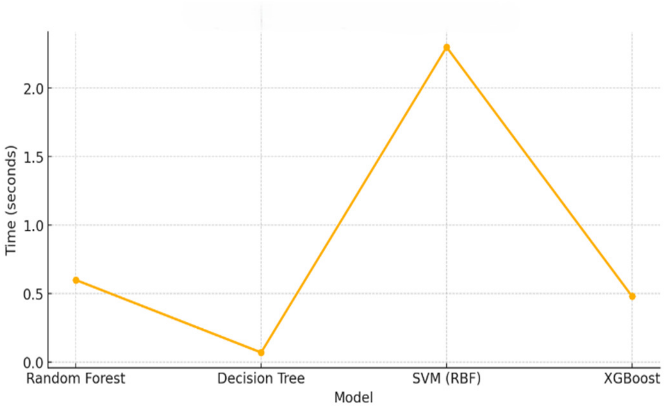

Fig. 7. Comparison of training times.

IV. CONCLUSION AND FUTURE WORK

A comparative evaluation of the four machine learning methods yields valuable insights into their effectiveness in detecting phishing URLs. Among the tested models, Extreme Gradient Boosting (XGBoost) demonstrated the highest performance across all key metrics while maintaining competitive training efficiency. The Random Forest (RF) model also showed strong results, achieving an impressive accuracy of 99.33% in detecting malicious traffic. Support Vector Machines (SVMs) achieved excellent accuracy but exhibited limitations in computational efficiency for real-time detection systems. In contrast, Decision Trees (DT), while the fastest to train, exhibited relatively lower overall effectiveness, suggesting they are more appropriate for rapid prototyping or use in resource-constrained environments rather than for deployment in production-level security systems. Overall, the findings emphasize the trade-off between detection accuracy and computational overhead. Ensemble classifiers such as XGBoost and RF appear to offer the most effective solutions for implementing phishing detection toolkits.

Future research could enhance model robustness by expanding the dataset with diverse phishing methods and emerging URL patterns. This would help improve generalization and increase the model's reliability. A detailed explanation of the dataset and model training procedures has been provided to ensure the reproducibility of results. By using tools such as scikit-learn and the XGBoost package, along with the dataset hosted on Kaggle, researchers are encouraged to replicate this work. The methodology provided serves as a robust basis for ongoing research and improvements in phishing detection.

REFERENCES

[1] M. A. Taha, H. D. A. Jabar, and W. K. Mohammed, "A Machine Learning Algorithms for Detecting Phishing Websites: A Comparative Study," *Iraqi Journal for Computer Science and Mathematics*, vol. 5, no. 3, Jan. 2024, Art. no. 13, https://doi.org/10.52866/ijcsm.2024.05.03.015.

[2] A. Bhavsar *et al.*, "Enhanced Phishing Website Detection: Leveraging Random Forest and XGBoost Algorithms with Hybrid Features," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 7, pp. 615-618, Jul. 2023.

[3] V.C. Kalyan, B.V.V. Satyanarayana, A.V.V. Laxman, A.V.S. Amarnath, and G. Hariharan, "Improving online safety with machine learning-based phishing detection," *International Journal of Progressive Research in Engineering Management and Science (IJPREMS)*, vol. 5, no. 4, pp. 1582–1587, Apr. 2025.

[4] M. Salem Alzboon, M. Subhi Al-Batah, M. Alqaraleh, F. Alzboon, and L. Alzboon, "Guardians of the Web: Harnessing Machine Learning to Combat Phishing Attacks," *Gamification and Augmented Reality*, vol. 3, Jan. 2025, Art. no. 91, https://doi.org/10.56294/gr202591.

[5] M. S. Islam, Mst. N. J. Jyoti, Md. S. Mia, and M. G. Hussain, "Fake Website Detection Using Machine Learning Algorithms," in *2023 International Conference on Digital Applications, Transformation & Economy (ICDATE)*, July 2023, pp. 255–259, https://doi.org/10.1109/ICDATE58146.2023.10248584.

[6] A. Mishra and Fancy, "Efficient Detection of Phising Hyperlinks using Machine Learning," *International Journal on Cybernetics & Informatics*, vol. 10, no. 2, pp. 23–33, May 2021, https://doi.org/10.5121/ijci.2021.100204.

[7] M. R. Islam, M. M. Islam, M. S. Afrin, A. Antara, N. Tabassum, and A. Amin, "PhishGuard: A Convolutional Neural Network Based Model for Detecting Phishing URLs with Explainability Analysis." arXiv, Apr. 27, 2024, https://doi.org/10.48550/arXiv.2404.17960.

[8] M. Elsadig *et al.*, "Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction," *Electronics*, vol. 11, no. 22, Jan. 2022, Art. no. 3647, https://doi.org/10.3390/electronics11223647.

[9] A. Fajar, S. Yazid, and I. Budi, "Enhancing Phishing Detection through Feature Importance Analysis and Explainable AI: A Comparative Study of CatBoost, XGBoost, and EBM Models." arXiv, Nov. 11, 2024, https://doi.org/10.48550/arXiv.2411.06860.

[10] S. Garg and S.S.M. Imran, "Recognition of malicious URLs using machine learning," *Indian Scientific Journal of Research in Engineering and Management*, vol. 8, no. 8, pp. 1–4, Aug. 2024.

[11] *State of the Phish 2023 – France Report*. Proofpoint, 2023

[12] "Web page Phishing Detection Dataset." https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset.

[13] "Phishing URL EDA and modelling." https://kaggle.com/code/akashkr/phishing-url-eda-and-modelling.

[14] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, Mar. 2019, https://doi.org/10.1016/j.eswa.2018.09.029.

[15] M. Dewis and T. Viana, "Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails," *Applied System Innovation*, vol. 5, no. 4, Aug. 2022, Art. no. 73, https://doi.org/10.3390/asi5040073.

[16] E. Y. Boateng and D. A. Abaye, "A Review of the Logistic Regression Model with Emphasis on Medical Research," *Journal of Data Analysis and Information Processing*, vol. 7, no. 4, pp. 190–207, Sept. 2019, https://doi.org/10.4236/jdaip.2019.74012.

[17] V. M. Yazhmozhi, B. Janet, and S. Reddy, "Anti-phishing System using LSTM and CNN," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, Aug. 2020, pp. 1–5, https://doi.org/10.1109/INOCON50539.2020.9298298.

[18] A. I. Adler and A. Painsky, "Feature Importance in Gradient Boosting Trees with Cross-Validation Feature Selection," *Entropy*, vol. 24, no. 5, May 2022, Art. no. 687, https://doi.org/10.3390/e24050687.

[19] M. S. K. Swaroop, K. R. Chowdary and S. Kavishree, "Phishing websites detection using machine learning," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 1470–1474, Apr. 2021.

[20] P. A. Bhavani, M. Chalamala, P. S. Likhitha, and C. P. S. Sai, "Phishing Websites Detection Using Machine Learning," Sept. 2022, https://doi.org/10.2139/ssrn.4208185.

[21] A. A. Albishri and M. M. Dessouky, "A Comparative Analysis of Machine Learning Techniques for URL Phishing Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18495–18501, Dec. 2024, https://doi.org/10.48084/etasr.8920.

[22] D. K. Singh and M. Shrivastava, "Evolutionary Algorithm-based Feature Selection for an Intrusion Detection System," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7130–7134, June 2021, https://doi.org/10.48084/etasr.4149.