Journal of Information Security Research

DLINE JOURNALS

Print ISSN: 0976-4143
Online ISSN: 0976-4151

JISR 2025: 16 (3)
https://doi.org/10.6025/jisr/2025/16/3/98-109

Algorithms and Methods for Detection of Phishing Website: A Review

Arun Kumar R, Vaishnavi H, Anusha S, Nausheen K, Apeksha P
Department of Computer Science and Engineering
Tontadarya College of Engineering, Gadag. India
Affiliated to Visvesvaraya Technological university, Belagavi. India
{kumararun37@gmail.com}, {vaishnavihattikal@gmail.com},
{anushaprakashshet@gmail.com}, {nausheenkanavi03@gmail.com}
{apekshapatil4001@gmail.com}

ABSTRACT

Phishing websites pose a serious threat, prompting the use of machine learning for detection. Researchers employ algorithms like XGBoost, Gradient Boosting, Adaboost, SVM, and Random Forest, trained on datasets to discern patterns distinguishing phishing from legitimate sites. Feature extraction analyzes URL structure, domain age, and content, while deep learning automates this process, identifying complex patterns. These techniques exhibit high accuracy, surpassing rule-based methods. Ongoing research positions machine learning as a crucial tool against online phishing, showcasing its potential in bolstering cybersecurity.

Keywords: Machine Learning Techniques, XGBoost, Gradient Boosting, Adaboost, SVM, Random Forest Evaluation

Received: 19 January 205, Revised 3 May 2025, Accepted 21 May 2025

1. Introduction

Phishing attacks, where bad actors try to trick people in to giving away sensitive information, are a big problem. They use sneaky tactics like fake websites that look real. Detecting and stopping these phishing websites is super important. This research looks into different ways to do that. Attackers often make fake websites that look like real ones, like banks or online stores. One way to catch them is by using smart computer programs called machine learning algorithms. These programs learn from a bunch of examples of real and fake websites to figure out the tricks bad guys use. Another way is by using systems that keep a list of known bad websites. These systems use information from blacklists, reports from users, and other sources to spot and block phishing sites. Some computer programs also watch how people behave on websites to catch anything weird or suspicious.

This paper talks about why it's crucial to stop phishing attacks, explores different ways to find these fake websites, and shows how using smart computer programs and watchful systems can help keep people safe online. It's like having online security guards that learn from the bad guys' tricks to protect us.

2. Literature Review of Detection of Phishing Website

2.1 Phishing Detection System Through Hybrid Machine Learning Based on URL

Aim: The aim of the paper is to propose a phishing detection system using a hybrid machine learning approach based on URL attributes. It addresses the issue of phishing attacks, which are dangerous cybercrime on internet [1].

Methods: Phishing detection based on URLs, utilizes machine learning algorithms such as decision tree, linear regression, naive Bayes, random forest, gradient boosting machine, support vector classifier, K-neighbors classifier and novel hybrid model (LR+SVC+DT).Dataset Preprocessing is used to remove null values. A cross-validation technique with grid search hyper-parameter is used to ensure that the models work well on different data samples. Researchers also employed canopy feature selection to identify and focus on the most relevant features. Finally, they made predictions to categorize phishing URLs and evaluated how well the model performed using metrics like accuracy (how often it was correct), precision (how many predicted positives were actually positive), recall (how many actual positives were correctly predicted), specificity (how many actual negatives were correctly predicted), and F1-score (a balance between precision and recall).

Conclusion: The paper concludes that a hybrid machine learning model (LR+SVC+DT) with canopy feature selection and grid search hyperparameter tuning effectively detects phishing URLs, achieving high accuracy, precision, recall, specificity, and F1-score. The proposed method achieves 98.12% accuracy and this approach, utilizing a unique combination of algorithms, distinguishes itself by showcasing improved performance metrics through a comprehensive methodology, including feature engineering and ensemble learning techniques.

2.2 Phishing URL Detection Using Machine Learning Methods

Aim: The aim of the research paper titled "Phishing URL detection using machine learning methods" is to investigate the linguistic and domain-based properties of URLs and create a machine learning model to detect fraudulent URLs used in phishing attacks [2].

Methods: The dataset comprises 3000 URLs, evenly split between malicious and benign categories, sourced from Phish Tank and the University of New Brunswick. After collection, data preprocessing involves merging and addressing null values. Feature extraction employs Python modules like urlparse and whois to extract lexical domain-based features. Subsequently, machine learning models, including Random Forest Classifier, Decision Tree, and Light GBM, are applied to the generated features for URL classification, incorporating both malicious and benign URLs in the analysis.

Conclusion: Among the machine learning models analyzed for identifying fraudulent URLs, Light GBM demonstrated superior performance with 89.5% training and 86% test accuracy. Following closely were Random Forest and Decision Tree, achieving 88.3% and 88% training, and 85.3% and 85% test accuracy, respectively. The feature importance graphs (Fig. 1) highlighted crucial factors influencing accuracy. Logistic Regression and SVM had slightly lower accuracies. In conclusion, the study underscores Light GBM as the most effective for detecting fraudulent URLs, with insights in to key features contributing to improved accuracy, providing valuable guidance for cybersecurity applications.

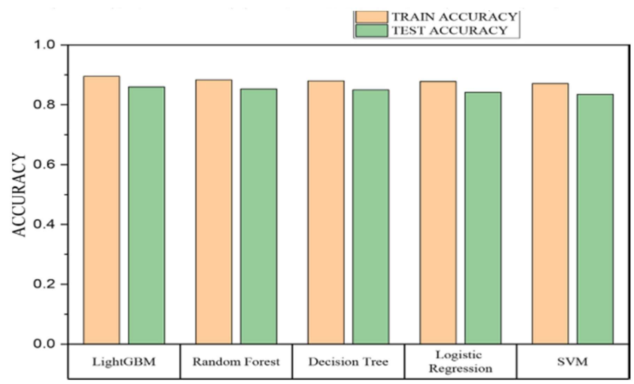

Figure 1. Accuracy scores for algorithms

2.3 Phishing URL Detection: A Real-Case ScenarioThrough Login URLs

Aim: The aim of the paper is to present a method for detecting phishing websites through URL analysis, using both machine learning and deep learning techniques [3].

Methods

2.3.1 Machine Learning Techniques: Feature extraction involves using handcrafted features proposed by Sahingoz et al. and statistical features using TFIDF combined with character N-gram. Handcrafted features include URL parsing using the tldextract library, extracting raw words, and evaluating various features like symbol frequency, domain randomness, subdomain count, etc. Eight supervised classifiers are employed: Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Random Forest (RF), Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Naïve Bayes (NB), and Logistic Regression (LR).

2.3.2 Deep Learning Techniques: Deep Learning involves using CNN models of Zhang et al. and Kim for URL classification. For machine learning, text classification includes text preprocessing, text representation (using TF-IDF combined with character N-gram), and a classifier. Features are extracted using the TF-IDF algorithm, operating on the character N-gram level.

Conclusion: The paper concludes by presenting PILU-90K, an updated dataset for phishing detection, highlighting the advantage of low false-positive rates. The recommended approach involves training models with legitimate login URLs, addressing issues of dependence on external services and ensuring real-world applicability, while identifying different categories of phishing attacks. Finally, we demonstrated that machine learning models using handcrafted URL features decreased their performance over time, up to 10:42% accuracy in the case of the LightGBM algorithm from the year 2016 to 2020. The conclusion emphasizes the importance of using recent data for machine learning methods and proposes future enhancements, including additional sample information and exploration of different URL codifications for improved detection performance.

2.4 A High-Accuracy Phishing Website Detection Method Based on Machine Learning

Aim: The aim of the paper is to develop a practical method to detect phishing websites based on machine learning approaches that provide high levels of accuracy and fast response times when detecting phishing attacks. The paper presents an improved predictive model that uses six different algorithms to enhance anti-phishing techniques. The experimental results show remarkable performances in detecting phishing websites [4].

Methods: The dataset contains information about phishing websites, totally 88,647 instances with 111 features. To better understand and analyze the data, visualization tools are employed to explore and explain the dataset. SMOTEENN technique is selected to address the challenges posed by imbalanced dataset in classification problems which also oversamples the minority class(phishing) and under sampling the majority class(non-phishing). After splitting the data in to training and testing sets, various models like Logistic Regression, KNN, Naïve Bayes, Random Forest, SVM and XGBoost are trained to recognize patterns and make predictions.

Conclusion: The study presents a robust phishing detection method using machine learning, achieving a remarkable 99.22% accuracy with the XGBoost classifier. The balanced dataset, feature selection, and efficient runtime make this approach a promising strategy for high-precision phishing website detection in online environments, surpassing the performance of other studies.

2.5 Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques

Aim: The paper aims to present a new way to detect fake websites that try to steal sensitive information like usernames, passwords, and credit card numbers. The new approach uses machine learning and looks at the website's URL and links to identify if it's a phishing website. The goal is to make the internet safer by detecting and stopping phishing attacks [5].

Methods

Feature Extraction: Analyzing URL structure for relevant phishing patterns. Extracting hyperlink-related features from the website's source code using a Document Object Model (DOM).

Rule Generation: Crucial step involving the creation of rules by analyzing URL structure and the DOM tree.

Hybrid Feature Set Creation: Combining features from URL and hyperlink analyses into an integrated set.

Machine Learning Classification: Training various machine learning classifiers with the hybrid feature set. Evaluating classifier performance based on criteria like error rate and accuracy.

Phishing Website Detection Algorithm: Outlining the process, likely involving feature extraction, rule application, model training, and classification for identifying and categorizing phishing or legitimate websites.

Conclusion: We made a way to catch phishing on your device by looking at the website address and its code. We tested it using different website addresses and got really good results—almost 99% accuracy in spotting tricky websites. Our method only looks at the address and code, which is a big success. While we think adding more features could make it even better, it might also make things more complicated. With everyone using mobile devices, we know that phishing on phones will be a big problem soon, so we're thinking about how to make our method work well for that too.

2.6 A Machine Learning Approach for Phishing Attack Detection

Aim: The aim of this research is to employ machine learning techniques to develop a robust methodology for accurately distinguishing between legitimate and phishing URLs. By leveraging various algorithms and performance evaluation metrics, the goal is to identify the most effective approach to enhance cybersecurity measures against phishing attacks [6].

Methods: Dataset Collection: Gathering comprehensive datasets from sources like PhishTank and UCI to incorporate diverse samples of both legitimate and phishing URLs. Feature Extraction: Identifying and extracting pertinent features from the URLs, such as domain age, syntax analysis, and URL length.

Algorithm Selection: Utilizing a range of machine learning algorithms including Extreme Gradient Boosting, Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine for classification purposes.

Training and Testing: Dividing the datasets into training and testing sets to train the models and evaluate their performance.

Performance Evaluation: Employing metrics such as accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) curves to assess the effectiveness of each model.

Hyper parameter Tuning: Fine-tuning the algorithms' parameters to optimize their performance.

Validation: Comparing the results obtained with existing methodologies or previous studies to validate the efficiency and reliability of the proposed approach.

Conclusion: In conclusion, this research underscores the substantial promise of employing machine learning algorithms, particularly Random Forest, in effectively distinguishing phishing URLs from legitimate ones. The developed methodology, leveraging diverse algorithms and comprehensive performance metrics, demonstrates robustness by achieving high accuracy rates, precision, recall, and AUCROC values. These findings signify a valuable contribution to cyber security, providing a proactive and reliable means to identify and mitigate the risks associated with phishing attacks. The results affirm the efficacy of this approach, solidifying its potential as an instrumental tool in fortifying digital defenses against evolving cyber threats.

2.7 Phishing Website Detection based on Effective Machine Learning Approach

Aim: The aim of the paper is to address the rising threat of phishing attacks by evaluating and comparing different techniques, with a focus on achieving higher accuracy in detection, ultimately identifying the Random Forest classifier as the most effective solution [7].

Methods: The paper explores the use of machine learning to identify phishing attacks and assesses their strengths and weaknesses. The authors introduce a Phishing Classification system with features designed to outsmart common detection methods. They employ numeric representation and compare traditional machine learning techniques like Random Forest, K Nearest Neighbors, Decision Tree, Linear SVC classifier, One- Class SVM classifier, and wrapper-based feature selection using URL metadata. The focus is on determining the legitimacy of websites, with particular emphasis on the effectiveness of the Random Forest algorithm.

Conclusion: Phishing is a growing threat, challenging to detect effectively. This study compares detection techniques, including Decision Tree and Random Forest. Random Forest outperforms others with 96.87% accuracy, notably addressing overfitting concerns. The research emphasizes its effectiveness in accurately identifying phishing websites, making it a preferred choice for enhanced cybersecurity.

2.8 Detection of Phishing Websites Using Machine Learning

Aim: The aim of this paper is to provide a comprehensive overview of the features and techniques used to detect phishing attacks using machine learning [8].

Methods

2.8.1 Random Forest Algorithm:

The Random Forest algorithm establishes a forest of decision trees, where a higher number of trees leads to increased detection accuracy. Trees are created through the bootstrap method, involving random selection with replacement of features and samples from the dataset. The algorithm then selects the optimal splitter for classification among the randomly chosen features, enhancing overall predictive performance.

2.8.2 Decision Tree Algorithm

The Decision Tree algorithm initiates by selecting the best attribute (splitter) for classification, serving as the root of the tree. It recursively constructs the tree until reaching a leaf node. The tree's internal nodes represent attributes, guiding decision making, while leaf nodes signify class labels, forming a training model for predicting target values or classes.

Conclusion: The paper highlights phishing threats and traditional detection methods' limitations. Using the Random Forest algorithm, a Chrome extension achieved 97.31% accuracy in phishing detection. Future plans involve building a scalable web service with online learning to adapt to new attack patterns and enhancing feature extraction for improved model accuracy, recognizing the importance of combining URL lexical features with others, such as host information.

2.9 Phishing Website Classification and Detection Using Machine

Aim: The aim of the paper is to compare different machine learning techniques for the phishing URL classification task and achieve the highest accuracy in detecting phishing websites [9]. Results of this article are highlighted in Table I.

Methods

- **Dataset Creation:**

1. Equal Representation: Balanced dataset with an equal number of labeled phishing and non phishing websites.

2. Random Mixing: Randomization to eliminate inherent order, ensuring unbiased representation.

- **Feature Extraction**

1. Lexical Structure: Features extracted from URL lexical structures using simple regular expressions.

- **Classifier Comparison:**

1. Various classifiers: Different classifiers employed for a comparative study.

2. Scikit-Learn: Utilization of Scikit-Learn for classification and predictive analysis.

- **Tool Usage**

1. Scikit-Learn: Employed for its robust classification and predictive analysis capabilities in the experiment.

Classifiers	Precis on	Reca ll	F1- Score	Accuracy in %
Logistic Regression	1	. 96	. 98	97.7
Random Forest	1	.96	.98	98.03
Gaussian Naïve Bayes	1	.95	.97	97.18
Decision Tree	1	.96	.98	98.02
KNearest Neighbor	.99	.97	.98	97.99

Table 1. Performance metrics results of different classifiers

Conclusion: The paper explores effective phishing URL classification, emphasizing dataset randomization, lexical analysis, and the impact of feature extraction on various classifiers. The study advocates continuous training with updated datasets for improved accuracy. Future enhancements may involve experimenting with additional features and incorporating rule-based content analysis to provide a more comprehensive phishing URL detection solution.

2.10 A Smart Model for Web Phishing Detection Based on New Proposed Feature Selection Technique.

Aim: Develop smarter web-phishing detection using novel feature selection, outperforming existing methods for accuracy [10].

Methods

Feature Evaluation and Ranking: A novel method is introduced to evaluate and rank diverse web content features, spanning URL attributes, HTML, JavaScript code, textual elements, images, and domain names.

Subset Extraction of Ranked Features: From the ranked features, an effective subset is extracted, chosen for its high classification accuracy in detecting web-phishing threats.

Construction of Random Forest Classifier: A Random Forest classifier is constructed in the third phase, trained using features from the previously identified subset.

Comparative Analysis of Feature Selection Methods: The new feature selection method is compared against established algorithms, aiming to demonstrate its superior performance in achieving higher classification accuracy comparative shown in Table 2.

Evaluation of Web-Phishing Detection Methodology: The effectiveness of the proposed web-phishing detection methodology is assessed, showcasing its superior performance in achieving the highest classification accuracy within the least possible time when compared to the adaptive Neuro-fuzzy inference system.

Classification Technique	Accuracy (%)
Naïve Bayes	92.15 %
K-Nearest Neighbors	96.16 %
Support Vector Machine	92.79 %
Random Forest	96.33 %
Artificial Neural Network	94.98 %

Table 2. Comparison between different classification techniques for web-phishing detection

Conclusion: This study created a smart way to catch tricky web-phishing, which tries to steal our information online. They used a three-step plan, inventing a super-smart feature checker and a powerful security system. Their new idea to pick the best features worked even better than the old ones. The security system they made works really fast, protecting us better than other systems. It's like a superhero for our online safety, making it a big step forward in keeping our information safe on the internet

3. Methodology

1. Start: Begin the process.

2. Register: Sign up or create an account.

3. Login: Log in with your account details.

4. Load Dataset: Bring in a bunch of data about websites, both good and potentially bad ones.

5. View Data: Take a look at the information in the dataset.

6. Model Training: Teach the computer to recognize patterns in the data to tell if a website is legit or phishing.

7. Prediction: Let the computer decide if a new website is likely a phishing site or not based on what it learned.

8. Graph: Create visuals like graphs to show how well the computer is doing in making these predictions.

9. Logout: Log out or exit the system when you're done.

This flowchart outlines the steps for a system that helps identify phishing websites using a trained computer model.

Dataset Collection

The first step of the research work was determining the right data set. The dataset selected was collected from Kaggle[11] for this task. The reasons behind selecting this dataset are several. It includes:

The data set is large, so working with it is intriguing

The number of features in the data set is 18 giving a wide range of features making the predictions a little more accurate.

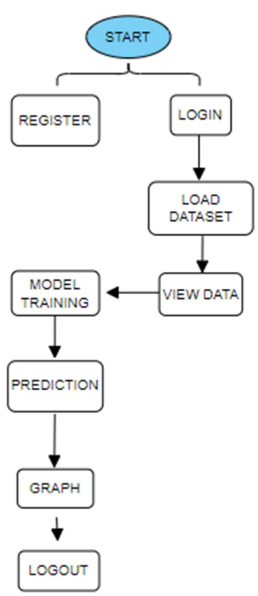

Figure 2. Work flow of proposed system

Algorithms

Random Forest: Random Forest is an ensemble learning method widely used in machine learning. It constructs multiple decision trees during training and merges them to enhance predictive accuracy and control overfitting. The "random" in its name comes from the process of selecting a random subset of features for each tree. This diversity ensures robustness and improves generalization to unseen data. Random Forest is versatile, excelling in classification and regression tasks, and is resilient to noisy datasets. Its popularity stems from high accuracy, ability to handle large datasets, and inherent feature importance assessment. These characteristics make it a go- to choice for various applications in predictive modeling.

XGBoost: Extreme Gradient Boosting, is a powerful machine learning algorithm renowned for its efficiency and predictive accuracy. Developed to enhance traditional gradient boosting, XGBoost utilizes a parallelized tree-building process and regularization techniques to prevent overfitting. It excels in various tasks, including classification, regression, and ranking. Its popularity stems from its speed, scalability, and ability to handle large datasets. XGBoost's key features include the incorporation of decision trees, handling missing data, and flexibility in customizing objective functions. Widely used in data science competitions, XGBoost has become a staple in machine learning pipelines for its exceptional performance and versatility.

Gradient Boosting: Gradient Boosting is a powerful machine learning ensemble technique that builds a strong predictive model by combining the predictions of multiple weak models, typically decision trees. It operates iteratively, focusing on minimizing errors by adjusting the weights of misclassified instances. Popular implementations include XGBoost, LightGBM, and AdaBoost. Gradient Boosting excels in various tasks, such as regression and classification, by sequentially adding models that correct the errors of their predecessors. This iterative learning process, coupled with the ability to handle complex relationships in data, makes Gradient Boosting a versatile and widely adopted method in the field of machine learning for achieving high predictive accuracy.

AdaBoost: Adaptive Boosting, is an ensemble learning technique in machine learning. It combines the strength of multiple weak learners to create a robust and accurate predictive model. Weak learners, often simple decision trees, are sequentially trained with a focus on correcting the errors made by their predecessors. During each iteration, misclassified instances are given higher weights, emphasizing their importance in subsequent models. AdaBoost adjusts the final prediction by aggregating the weighted votes of all weak learners. Its adaptability to diverse data and ability to handle complex problems make AdaBoost a popular choice for classification tasks in various domains.

SVM: Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates data points into different classes, maximizing the margin between them. SVM is effective in high-dimensional spaces and is capable of handling non-linear relationships through the use of kernel functions. It is widely used in various fields such as image recognition, text classification, and bioinformatics. SVM aims to achieve optimal generalization performance by minimizing classification errors while maintaining a maximum margin between decision boundaries. Its robustness and versatility make it a popular choice in machine learning applications.

Voting Classifier: The Voting Classifier is an ensemble machine learning model that combines the predictions of multiple base classifiers to improve overall accuracy and robustness. It operates by aggregating the individual predictions of its constituent classifiers and selecting the most common or averaged outcome. The three main types of voting classifiers are hard voting, soft voting, and stacking. In hard voting, the majority vote determines the final prediction, while soft voting considers the weighted average of class probabilities.

This approach is particularly effective in enhancing model performance, mitigating overfitting, and addressing diverse data patterns, making it a popular choice in various classification tasks.

4. Results

Among the machine learning algorithms for detecting fraud websites, it is found that Random Forest gives the best result with accuracy of 83% compared to other machine learning algorithms.

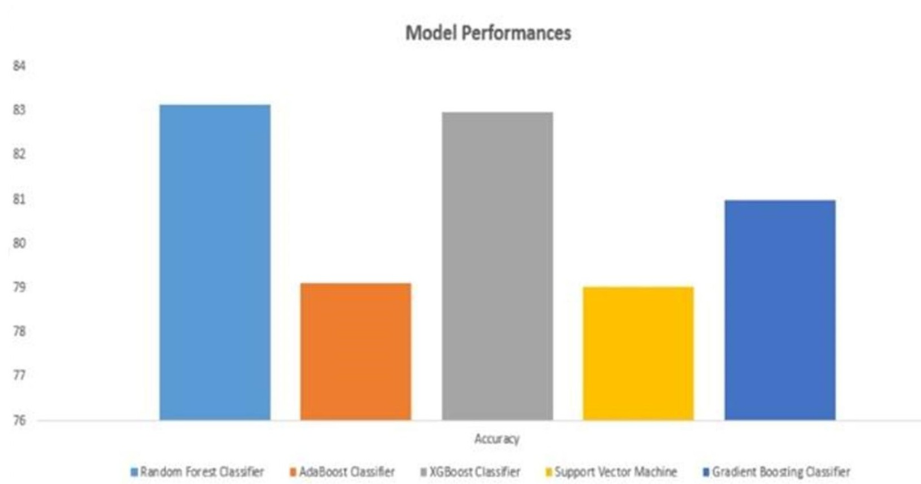

Figure 3. Graph representing model performance with respect to accuracy

5. Conclusion

In conclusion, using smart computer programs that learn from lots of data helps us find and stop fake Websites that try to steal our personal information. These programs, like XGBoost and others, look at different parts of a website, like its address or how old it is, to figure out if it's trying to trick us. They've shown they can be really good at this, doing better than older methods. So, these smart programs are becoming important tools to protect us from online scams, and as researchers keep working on them, they'll likely get even better, making the internet a safer place for everyone.

References

[1] Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on URL. *IEEE Access, 11*, 36805–36822. https://doi.org/10.1109/ACCESS.2023.3252366

[2] Ahammad, S. H., Islam, M. R., Hassan, M. M., Kabir, M. N., Rahman, M. M., Alhumyani, H. (2022). Phishing URL detection using machine learning methods. *Advances in Engineering Software, 173*, 103288. https://doi.org/10.1016/j.advengsoft.2022.103288

[3] Sanchez-Paniagua, M., Fernandez, E. F., Alegre, E., Al-Nabki, W., Gonzalez-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. *IEEE Access,* 10, 42949–42960. https://doi.org/10.1109/ACCESS.2022.3168681

[4] Bahaghighat, M., Ghasemi, M., Ozen, F. (2023). A high-accuracy phishing website detection method based on machine learning. *Journal of Information Security and Applications,* 77, 103553. https://doi.org/10.1016/j.jisa.2023.103553

[5] Das Guptta, S., Shahriar, K. T., Alqahtani, H., Alsalman, D., Sarker, I. H. (2022). Modeling hybrid feature-based phishing websites detection using machine learning techniques. *Annals of Data Science*. https://doi.org/10.1007/s40745-022-00379-8

[6] Choudhary, T., Mhapankar, S., Bhddha, R., Kharuk, A., Patil, R. (2023). Machine learning approach for phishing attack detection. *Journal of Artificial Intelligence and Technology*. https://doi.org/10.37965/jait.2023.0197

[7] Harinahalli Lokesh, G., Bore Gowda, G. (2020). Phishing website detection based on effective machine learning approach. *Journal of Cyber Security Technology,* 5(1), 1–14. https://doi.org/10.1080/23742917.2020.1813396

[8] Borde, S., Deshpande, A. (2021). Detection of phishing websites using machine learning. *[Unpublished manuscript or technical report].*

[9] Kumar, J., Santhanavijayan, A., Bs, B., Janet, B. (2020). Phishing website classification and detection using machine learning. *[Conference paper or unpublished work].*

[10] El-Rashidy, M. (2020). A smart model for web phishing detection based on new proposed feature selection technique. *Menoufia Journal of Electronic Engineering Research,* https://doi.org/10.21608/mjeer.2020.32404.1021

[11] Kaggle. *Phishing website detection CSV dataset*. https://www.kaggle.com/datasets/csepython/phishing-website-detection-csv-dataset