*Vol. 3, No. 3, pp 16 - 29, 2022*
*DOI: 10.55969/paradigmplus.v3n3a2*

PARADIGM Plus

# Hybrid Approach for Phishing Website Detection Using Classification Algorithms

Mukta Mithra Raj[1] ✉ and J. Angel Arul Jothi[1]

[1]*Birla Institute of Technology and Science Pilani, United Arab Emirates*
f20190134@dubai.bits-pilani.ac.in, angeljothi@dubai.bits-pilani.ac.in

**Abstract**

The internet has significantly altered how we work and interact with one another. Statistics show 63.1% of the present world population are internet users. This clearly indicates how heavily man is dependent on digital media. Digital media users are on the rise and so is the incidence of cybercrimes. People who lack experience and knowledge are more vulnerable and susceptible to phishing scams. The victims experience severe consequences as their personal credentials are at stake. Phishers use publicly available sources to acquire details about the victim's professional and personal history. Countermeasures must be implemented with the highest priority. Detection of malicious websites can significantly reduce the risk of phishing attempts. In this research, a highly accurate website phishing detection method based on URL features is proposed. We investigated eight existing machine learning classification techniques for this, including extreme gradient boosting (XGBoost), random forest (RF), adaptive boosting (AdaBoost), decision trees (DT), K-nearest neighbors (KNN), support vector machines (SVM), logistic regression and Naive Bayes (NB) to detect malicious websites. The results show that XGboost had the best accuracy with a score of 96.71%, followed by random forest and AdaBoost.We further experimented with various combinations of the top three classifiers and observed that XGboost-Random Forest hybrid algorithms produced the best results. The hybrid model classified the websites as legitimate or phishing with an accuracy of 97.07%

***Keywords:*** URL Features · Data Mining · Machine Learning · Hybrid Classification Algorithms · Phishing Website Detection

*Received: 27 October 2022 · Accepted: 16 December 2022 · Published: 20 December 2022.*

## 1 Introduction

Website phishing [1] is the unethical practice of designing websites that appear genuine and legitimate but are actually created to gain unauthorized access and steal information. A malicious website attempts to deceive people into revealing private information. It accomplishes this by making you believe that the website is genuine. For many people, the internet serves as a window to the outside world. Society's reliance on the digital world has increased during the pandemic. They use it for a variety of tasks, including storing data and conducting online financial transactions. Consequently, there was an increase in phishing websites and many people fell prey to phishing attacks. Unknowingly, the vulnerable customers would attempt to click on the fake URLs that appeared cloned and genuine. They are stripped of their personal information and login details by such harmful phishing

attacks. Customers and businesses are both at risk, which might result in major financial loss and business interruption.

Eighty-five percent of phishing attacks are aimed at stealing user credentials, according to IBM's 2021 [2] data breach report. This will result in additional intrusions and cybercriminals can access files and observe user behavior. There could be severe and even severe consequences for a business, its staff, and its clients. Prevention is the key to altering disastrous consequences. A cutting-edge automatic phishing detection system is required to counter dangerous threats. Models incorporated into web browsers can identify phishing activities. Automated phishing detection algorithms can become more efficient if website properties are integrated with the input dataset from a large number of websites.

Machine Learning classification algorithms can detect phishing websites efficiently in less time. Compared to the deep learning method, it works effectively with less training data and is much faster. Another advantage of machine learning over deep learning is that It does not require specific hardware like Graphics Processing Units (GPU) for implementation. Therefore, the objective of this research is to apply machine learning classification algorithms for the accurate identification of phishing websites. Eight different classification algorithms are used and tested on the dataset, including extreme gradient boosting (XGBoost), random forest (RF), adaptive boosting (AdaBoost), decision trees (DT), K-nearest neighbors (KNN), support vector machines (SVM), logistic regression and Naive Bayes (NB) to identify malicious URLs. The results are compared, and the best three in terms of accuracy are utilized to construct hybrid algorithms. The research also explores the impact of using fewer features on the classification performance of the classifiers.

The research report is structured as follows: Related work is included in Section 2. The dataset is described in Section 3 of the document. Section 4 details the approach and the classification algorithms. Implementation details are given in Section 5. The findings and discussion are presented in Section 6. Section 7 concludes the research paper with a scope of future work.

## 2 Related Works

This section includes a brief overview of the ensemble approaches presented in some of the recent works.

An ensemble model was presented by Jiaqi Gu and Hui Xu [3]. The proposed model is developed by integrating XGBoost with KNN and RF as base learners. When compared with other ensemble models, the XGBoost ensemble model achieved better run time and an accuracy of 96.44%. Further, it can safeguard internet users when embedded into an application or web extension.

In the proposed work by Maini et al. [4] URL features were tested using eight classification algorithms, namely, XGBoost, RF, AdaBoost, DT, KNN, SVM, logistic regression, and NB. XGBoost attained the highest accuracy of 93.2%. The voting classifier is then used to construct the ensemble model. According to the results, the suggested ensemble technique performed better than XGBoost, with an accuracy of 93.6%.

An efficient random forest-support vector machine (RF-SVM) hybrid model was developed by Pandey et al. [5]. According to the experimental results, the accuracy of the RF-SVM hybrid model is superior to that of the RF and SVM. The dataset is divided into multiple parts and trained using random forest. Each subunit is bagged to create the final decision unit. To improve accuracy, the RF subunits are then categorized using SVM. The SVM and RF results are bagged to merge the results. The hybrid model can detect phishing websites with a high rate of accuracy.

In a similar study, Ramana et al. [6] developed an ensemble of XGBoost, RF, and DT based on feature selection to classify phishing websites. Experiments were conducted using UCI and Mendeley datasets. The trial with the former achieved an accuracy of 97.51% while that with the latter scored 98.45%. The final model was implemented using wrapper feature selection techniques and filters.

The efficiency of ensemble models in detecting phishing websites is also evident in the SVM, decision tree, and random forest ensemble model proposed by Abusaimeh et al. [7]. The proposed model was compared with each classifier and thereafter implemented and analyzed using the dataset. The results demonstrate that the ensemble model outperforms the three algorithms.

The hybrid framework developed by Tabassum et al. [8] based on merging bagging and boosting techniques was built using SVM, DT, RF, and XGBoost. The URL features were selected and to achieve better results, the dimensionality of the feature subset was reduced. An accuracy of 98.28% was obtained using the hybrid approach.

Another ensemble method was devised by Lakshmanarao et al. [9] by combining RF, DT, and Gradient Boosting. A variety of classification techniques were tested utilizing the dataset after preprocessing. Later, two sets of priority algorithms were created grouping the individual classifiers. The best-performing algorithms from the two sets were combined to construct the final ensemble algorithm. The proposed model achieved an accuracy of 97%.

The ensemble framework suggested by Subasi et al. [10] consists of SVM and AdaBoost classification algorithms for the detection of phishing websites. The approach improved accuracy, F1 score, and ROC curve. According to the results obtained, the ensemble model can be used to detect phishing pages with a success rate of 97.61 percent.

Zamir et al, [11] proposed two stacking models. The first model combines bagging, neural networks, and random forests. The second model consists of random forest, KNN, and bagging. Principal component analysis (PCA) was used to increase the proposed models' accuracy. The first stacking model achieved the highest accuracy of 97.4% when compared to the individual classifiers and the second stacking model.

Kalabarige et al. [12] proposed a multilayered stacked ensemble learning technique (MLSELM ) using the multi-layer perceptron (MLP), XGB, KNN, RF, and LR at various layers to improve performance. Four distinct datasets were used to evaluate the model, and the accuracy ranged from 96.79% to 98.90%. The proposed model performed better with a balanced dataset than with an imbalanced dataset.

The machine learning approach formulated by Makkar et al. [13] evaluated bagged AdaBoost, bayesian generalized linear model, Naïve Bayes, linear SVM with class weights, ensembles of generalized linear models, monotone multilayer perceptron neural network, quadratic discriminant analysis, multilayer perceptron, neural networks with feature extraction, and oblique RF. Following that, 10 rounds of cross-validation are performed on an ensemble of the best three best in terms of accuracy. A total accuracy of 97.27% is shown by the tested models.

Yang et al. [14] proposed an ensemble method using RF and convolutional neural networks (CNN) to predict the authenticity of websites. The integration of CNN improved the performance of the model. URLs are transformed into fixed-size matrices using character embedding strategies. The proposed model recorded a 99.35% accuracy rate.

In the study by Al-Sarem et al. [15] an improved stacking ensemble technique for phishing website detection was proposed. Random forests, AdaBoost, XGBoost, Bagging, GradientBoost, and LightGBM were among the ensemble machine learning techniques whose parameters were optimized using a genetic algorithm (GA). The top three models were used to build the ensemble classifiers. According to the experimental findings, the proposed optimal stacking ensemble method achieved an accuracy of 97.16%.

A voting ensemble model, an Expandable Random Gradient Stacked Voting Classifier ( ERG-SVC) was proposed by Indrasiri et al. [16] after experimenting with seven classifiers. 22 best features were selected after feature selection and the trials were conducted with different datasets. With a prediction accuracy of 98.118%, GB surpassed other algorithms individually while the ensemble model achieved an accuracy of 98.27%.

Alsaedi et al. [17] proposed an ensemble model with RF and MLP. RF is used for pre-classification while MLP is used for decision-making. To enhance detection performance, features based on cyber threat intelligence (CTI) are applied. Compared to the conventional URL-based model, the proposed

CTI-based detection model has an improved accuracy of 7.8%.

The goal of this study is to recommend a hybrid classification model. Thus, in our work, we analyzed and considered eight different machine learning classifiers applied in the publications. The accuracy rate and F1 score are measured using 30, 15, 10, and 5 URL features.

## 3 Dataset Description

The Kaggle[1] dataset for phishing websites is utilized in this work. It includes the URLs of more than 11,000 websites. Each URL has 30 attributes. Values 1 and -1 are used to classify the websites as legitimate and phishing respectively. Table 1 lists the parameters of a URL, its types, and its values. Values range from two to three which represent an attribute's range of strength from low to high.

Table 1: URL Attributes, Types, and Values

| Attribute | Attribute type | Value |
|---|---|---|
| UsingIP, ShortURL, Symbol@, Redirecting, PrefixSuffix, DomainRegLen, Favicon, NonStdPort, HTTPSDomainURL, RequestURL, InfoEmail, AbnormalURL, StatusBarCust, DisableRightClick, UsingPopupWindow, IframeRedirection, AgeofDomain, DNSRecording, PageRank, GoogleIndex, StatsReport | Categorical | {-1,1} |
| LongURL, SubDomains, HTTPS, AnchorURL, LinksInScriptTags, ServerFormHandler, WebsiteTraffic, LinksPointingToPage | Categorical | {-1,0,1} |
| WebsiteForwarding | Categorical | {0,1} |

## 4 Methodology

Phishing websites can be identified by analyzing the URL features of the website. Figure 1 illustrates the method used in this work to classify the websites. Definite features that constitute each URL are selected. The selected features are then utilized for training and testing in which 70% of the samples in the dataset are used for training and 30% for testing. The classifiers and hybrid models are then trained to determine whether a website is phishing or not.

[1] https://www.kaggle.com/code/eswarchandt/website-phishing/data

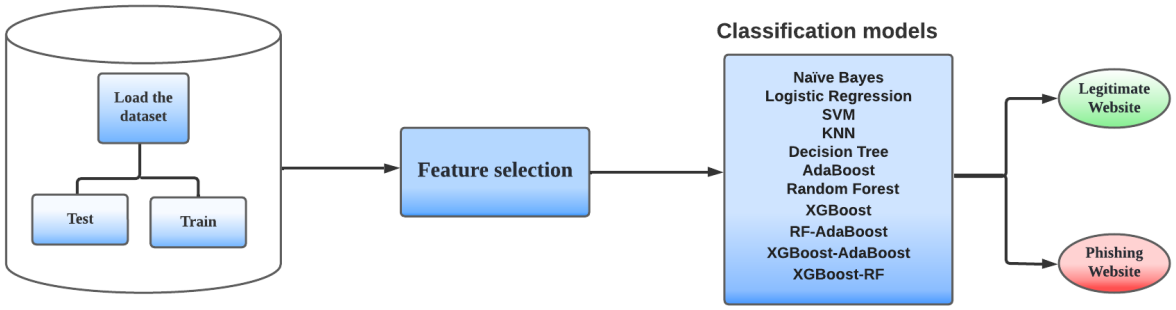


Figure 1: Proposed architecture.

The algorithms applied for the classification of the websites are extreme gradient boosting (XGBoost), Random Forest, Support Vector Machine, Naive Bayes, logistic regression, K-Nearest Neighbours, decision tree, and AdaBoost. The following subsections explain these classification algorithms.

## 4.1 Machine Learning Algorithms

### 4.1.1 XGBoost

XGBoost is typically used for regression and classification on large datasets. Using sequentially built shallow decision trees and a highly scalable training method that reduces overfitting, it produces accurate results. The learning rate, number of trees, percentage of randomly sampled columns, gamma, and maximum depth per tree are among the significant hyperparameters. As performance improves, complexity and overfitting also increase. Overfitting is enhanced by the column fraction. Regularization depends on the value of the regularization parameter gamma, which increases as the gamma value increases [18].

### 4.1.2 Random Forest (RF)

The RF classifier algorithm uses a collection of distinct decision trees that have not been pruned and are fully developed. This algorithm uses randomness to generate each of the separate trees, which are then combined to generate a prediction. For large datasets, this approach is more effective than a single decision tree and creates far less variance. It almost exactly approximates missing data. The number of trees, the minimum number of data objects required to divide an internal node, the minimum number of samples needed to be present in the leaf node, and the number of jobs that run simultaneously are the hyperparameters used [19].

### 4.1.3 Support Vector Machine (SVM)

The SVM algorithm can be applied to classification or regression tasks. The key component of it is a hyperplane that splits the attribute space into two groups: legitimate or phished websites. The classification of the websites is accomplished by placing the legitimate websites on one side of the plane and the phished websites on the other once the algorithm has been trained using the dataset. Potential errors caused by overtraining are lessened as a result. The kernel hyperparameter and the probability hyperparameter specify the type of algorithm to be used to enable the probability estimates [20].

### 4.1.4 Naive Bayes (NB)

The NB classification algorithm gives predictions based on the probability of an object. In this study, an NB classifier called Bernoulli Naive Bayes was used, and the features comprised boolean variables. Features are regarded as independent of one another. The only parameter that cannot be modified is the number of classes [20].

### 4.1.5 Logistic Regression

Logistic regression is used for both classification and regression. It is widely used for binary classification problems. LR uses the sigmoid function to determine a label's probability [20].

### 4.1.6 K-Nearest Neighbors (KNN)

The KNN approach is primarily employed for classification although it can be utilized for regression issues. It categorizes two data objects using a similarity or distance metric. K represents the number of neighbors and the value of K in this work is 5. Among the crucial hyperparameters are the number

of neighbors, the distance metrics used to determine the neighborhood's composition, and the weight function for prediction [18].

#### 4.1.7 Decision Tree (DT)

DT constructs a tree structure to address classification and regression problems. The attribute with the highest information gain is taken as the splitting attribute or the decision node. The class label is represented by the leaf nodes in the decision tree. The stopping criterion of a decision tree is when the child node has a homogeneous class. The number of trees is the most important hyperparameter used [18].

#### 4.1.8 AdaBoost

AdaBoost, an ensemble technique, can boost the performance of any machine learning algorithm. It works best when employed with weak learners. With a learning rate of 1, the AdaBoost used here has 50 decision trees. Selecting a weak learner to train the model, the number of weak learners to train, and the weights of the weak learners that affect the learning rate are the essential hyperparameters [10].

### 4.2 Hybrid Model

A hybrid model is constructed to improve the accuracy of the classification model. The voting classifier is used to create the hybrid classifiers. It is a machine learning model that is used to train an ensemble of algorithms to carry out classification tasks. Hybrid models of XGBoost and RF, XGBoost and AdaBoost, and RF and Adaboost are created using the voting classifier. The hybrid models are then trained using the same dataset, and the XGBoost-RF hybrid classification model achieved the highest testing accuracy of 97.07%.

## 5 Implementation

This work was developed using the Jupyter Notebook accessed from Anaconda with Python version 3.10.4. The in-built machine learning library scikit-learn was utilized to build the eight classification models. The models were built by using 70% of the dataset for training and the rest 30% for testing. The confusion matrix is formed by considering the samples that fall into the predicted and actual classes and is used to formulate the evaluation metrics. The evaluation metrics used to compare the algorithms are accuracy, error rate, recall, specificity, precision, and F1 score given in Table 2. These metrics are calculated using True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). When a legitimate website is correctly predicted as legitimate it is TP, whereas if the prediction is false, then it is FN. When a phishing website is rightly identified as phishing, it is TN, while if it is incorrect, it is FP.

Table 2: Evaluation Metrics

| | |
|---|---|
| True Positive Rate (TPR) / Recall /Sensitivity | TP / (TP+FN) |
| True Negative Rate (TNR) /Specificity | TN / (FP+TN) |
| Accuracy | (TP+TN) / (TP+FP+TN+FN) |
| Error Rate | (1 - Accuracy) |
| Precision | TP/ (TP+FP) |
| F1 score | 2 * Precision * Recall/ (Precision + Recall) |

## 6 Results and Discussion

An imbalanced dataset was used for training and testing the individual classifiers and the hybrid models. The significance of feature selection on the classification models was explored using different numbers of attributes.

The performance of all models was initially examined using each of the 30 attributes, with the exception of the "index" feature because it is not used to classify websites. The experiment was carried out again with the top 15, 10, and 5 attributes for the models displayed in Table 3. The average impurity decrease estimated from each decision tree was used for feature selection in AdaBoost, Random Forest, Decision Tree, K-Nearest Neighbors, XGBoost, XGBoost-Random Forest, XGBoost-AdaBoost, and Random Forest-AdaBoost [21]. In the same way, the dataset attributes were assigned coefficients to determine the feature selection in logistic regression and SVM. In Naive Bayes, the predictive value of a feature is evaluated by observing how the prediction error increases when a feature is missing.

Table 3: Attributes used

| Algorithms | 15 Attributes | 10 Attributes | 5 Attributes |
|---|---|---|---|
| AdaBoost, Decision Tree | HTTPS, AnchorURL, WebsiteTraffic, LinksInScriptTags, SubDomains, PrefixSuffix, ServerFormHandler, LinksPointigToPage, PageRank, RequesstURL, GoogleIndex, DomainRegLen, AgeofDomain, UsingIP, LongURL | HTTPS, AnchorURL, WebsiteTraffic, LinksInScriptTags, SubDomains, PrefixSuffix, ServerFormHandler, LinksPointigToPage, PageRank, RequesstURL | HTTPS, AnchorURL, WebsiteTraffic, LinksInScriptTags, SubDomains |
| KNN | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, LinksInScriptTags, PrefixSuffix, RequestURL, ServerFormHandler, LinksPointingToPage, AgeofDomain, DomainRegLen, UsingIP, GoogleIndex, DNSRecording, PageRank | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, LinksInScriptTags, PrefixSuffix, RequestURL, ServerFormHandler, LinksPointingToPage, AgeofDomain | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, LinksInScriptTags |

| | | | |
|---|---|---|---|
| Logistic Regression | AnchorURL, PrefixSuffix, HTTPS, WebsiteFowarding, ServerFormHandler, LinksPointingToPage, LinksInScriptTags, GoogleIndex, SubDomains, WebsiteTraffic, UsingIP, NonStdPort, HTTPSDomainURL, ShortURL, DNSRecording | AnchorURL, PrefixSuffix-, HTTPS, WebsiteFowarding, ServerFormHandler, LinksPointingToPage, LinksInScriptTags, GoogleIndex, SubDomains, WebsiteTraffic | AnchorURL, PrefixSuffix, HTTPS, WebsiteFowarding, ServerFormHandler |
| Naive Bayes | HTTPS, PrefixSuffix, AnchorURL, SubDomains, WebsiteTraffic, UsingIP, GoogleIndex, DNSRecording, AgeofDomain, PageRank, RequestURL, ServerFormHandler, LinksPointingToPage, HTTPSDomainURL, WebsiteForwarding | HTTPS, PrefixSuffix, AnchorURL, SubDomains, WebsiteTraffic, UsingIP, GoogleIndex, DNSRecording, AgeofDomain, PageRank | HTTPS, PrefixSuffix, AnchorURL, SubDomains, WebsiteTraffic |
| Random Forest, RF-AdaBoost | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, PrefixSuffix, LinksInScriptTags, RequestURL, ServerFormHandler, LinksPointingToPage, DomainRegLen, AgeofDomain, UsingIP, DNSRecording, GoogleIndex, PageRank | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, PrefixSuffix, LinksInScriptTags, RequestURL, ServerFormHandler, LinksPointingToPage, DomainRegLen | HTTPS, AnchorURL, WebsiteTraffic, SubDomains, PrefixSuffix |

| | | | |
|---|---|---|---|
| SVM | AnchorURL, PrefixSuffix, HTTPS, WebsiteFowarding, LinksPointingToPage, LinksInScriptTags, ServerFormHandler, SubDomains, GoogleIndex, UsingIP, ShortURL, NonStdPort, WebsiteTraffic, InfoEmail, HTTPSDomainURL, | AnchorURL, PrefixSuffix, HTTPS, WebsiteFowarding, LinksPointingToPage, LinksInScriptTags, ServerFormHandler, SubDomains, GoogleIndex, UsingIP | AnchorURL, PrefixSuffix, HTTPS, WebsiteFowarding, LinksPointingToPage |
| XGBoost, XGBoost-RF, XGBoost-AdaBoost | HTTPS, AnchorURL, PrefixSuffix, ServerFormHandler, WebsiteTraffic, LinksInScriptTags, SubDomains, GoogleIndex, ShortURL, DomainRegLen, PageRank, LinksPointingToPage, AgeofDomain, UsingIP, InfoEmail | HTTPS, AnchorURL, PrefixSuffix, ServerFormHandler, WebsiteTraffic, LinksInScriptTags, SubDomains, GoogleIndex, ShortURL, DomainRegLen | HTTPS, AnchorURL, PrefixSuffix, ServerFormHandler, WebsiteTraffic |

The performance of the classifiers for the various evaluation metrics for the attributes 30, 15, 10, and 5 are compared in Figures 2, 3, 4, and 5 respectively. The performance of RF was optimized by determining the ideal value for the number of decision trees created. It was achieved by determining the average predictions and estimating the number of data items predicted for a specific dataset sample accurately, where the set of random values ranges from 40 to 300. The number of decision trees that worked best for 30 attributes was 200, followed by 20 for 15 attributes, 280 for 10 attributes, and 80 for the five most crucial attributes.

From the results of Figure 2 for 30 attributes, XGBoost-Random Forest is the best classifier because it has the highest accuracy (97.07%) and F1 score (96.67%) among the classifiers. Due to its low accuracy and poor F1 score, the Naive Bayes classifier exhibits the least performance. When compared to XGBoost-AdaBoost, XGBoost-RF displays good accuracy and F1 score but a lower precision value. As compared to XGBoost-Random Forest, other classification models are found to have lesser accuracy, F1 scores, and recall values.

For the 15 attributes shown in Figure 3, Random Forest-AdaBoost has the highest accuracy (95.7%) and F1 score (95.44%), whereas the Naive Bayes classifier has the lowest. According to Figure 4 for 10 features, XGBoost-Random Forest and XGBoost-AdaBoost are at the top of the list with accuracy rates of 94.96% and F1 scores of 94.31 percent, respectively, while Naive Bayes performs the worst.

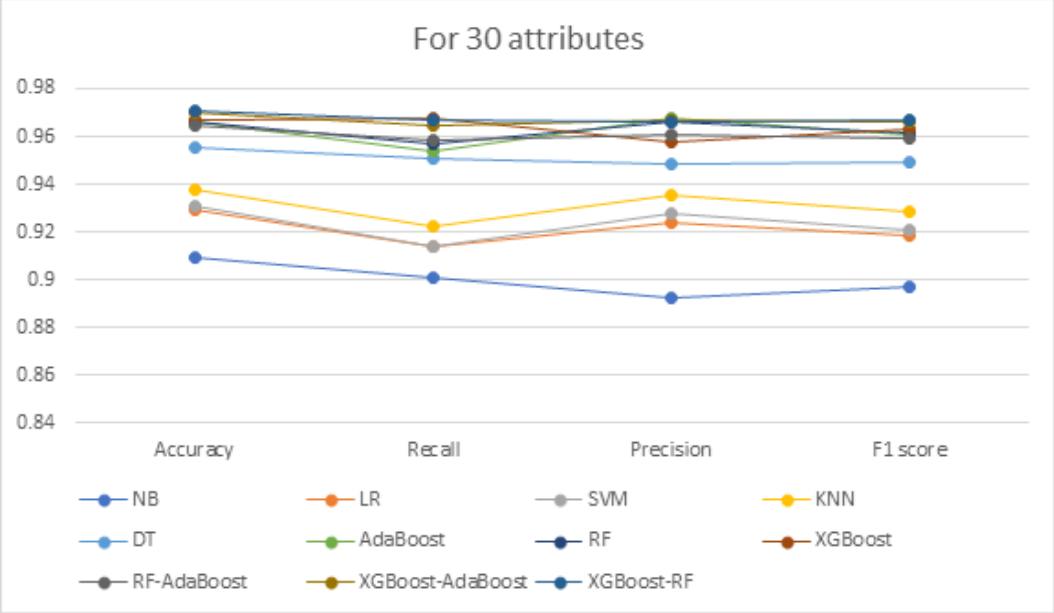


Figure 2: Comparison of evaluation metric values for 30 attributes.

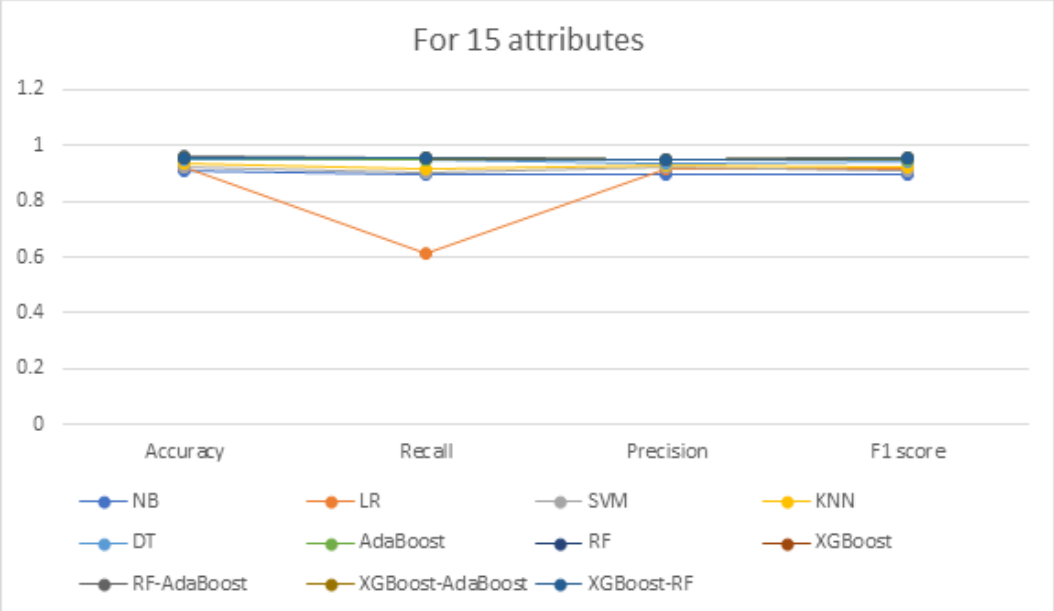


Figure 3: Comparison of evaluation metric values for 15 attributes.

AdaBoost has the best accuracy of 93.03%, while RF-AdaBoost has the highest F1 score of 92.11% for the five attributes represented in Figure 5. Compared to the other models, SVM ranks the lowest.

The results from Tables 3 showed that XGBoost-Random Forest performed the best. Additionally, the performance of the classification was not enhanced by feature selection. The efficiency of the classifiers increases with more attributes and decreases substantially as the number of attributes is decreased. The dataset used does not contain any noise or irrelevant data and hence all attributes are

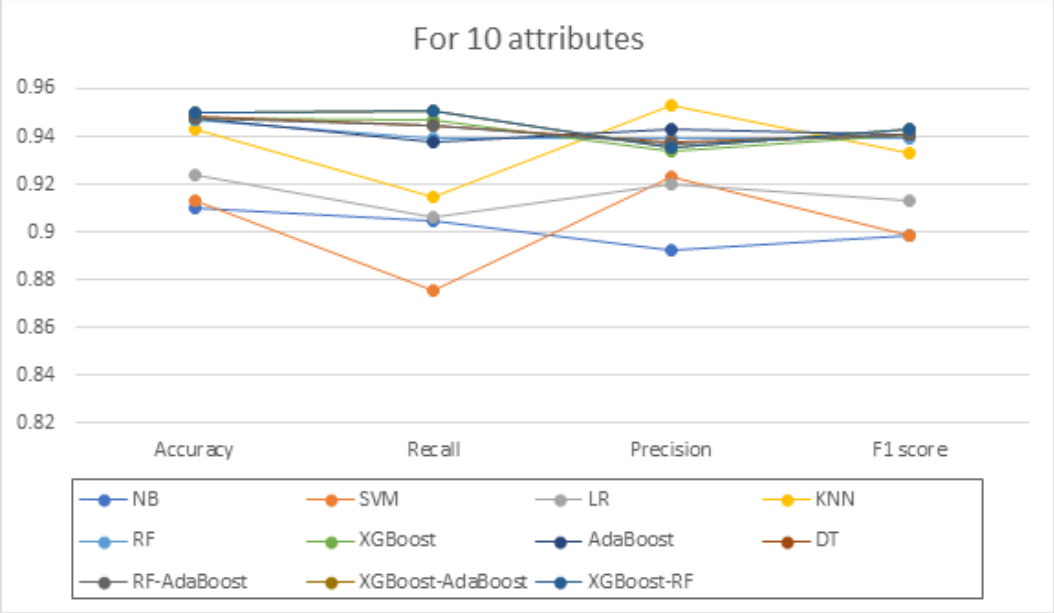


Figure 4: Comparison of evaluation metric values for 10 attributes.

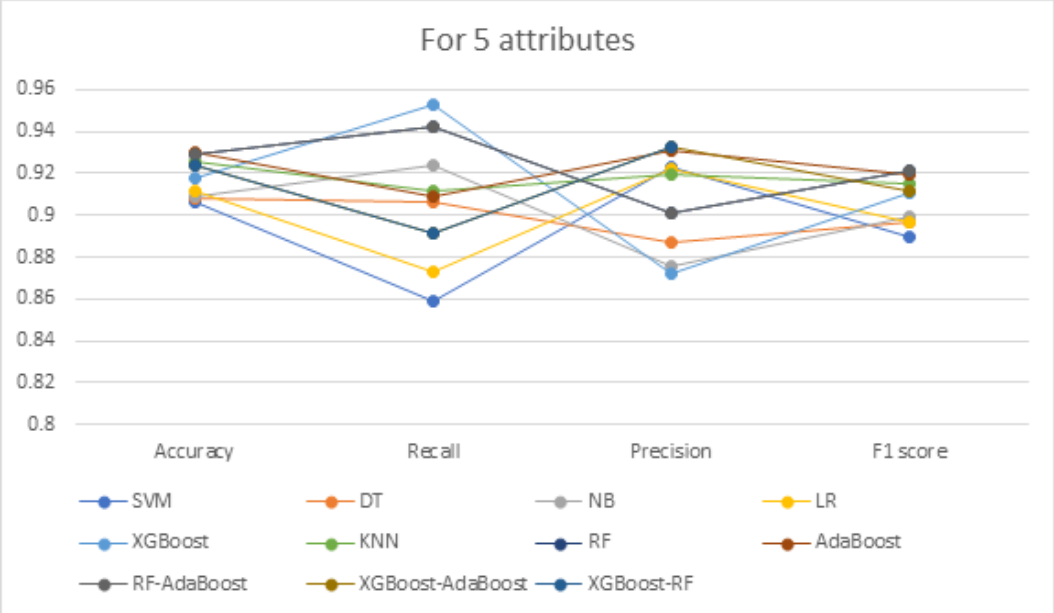


Figure 5: Comparison of evaluation metric values for 5 attributes.

considered to obtain a good result. The quality of the attributes in the dataset is more important than their quantity in order to get better results.

The comparison of findings from this research and previous studies [3, 9] are given in Table 4. [3] uses the same dataset as this study. An ensemble model comprising XGBoost, RF, and KNN is proposed in [3]. The outcomes of [3] demonstrate that the ensemble model outscored all other classifiers, attaining an accuracy and F1 score of 96.43% .

Similar datasets are used in this research and [9], except for one record. [9] utilized a dataset[2] that comprised 11,055 records, whereas the current study utilizes 11,054 records. The best performance was produced by [9] using a stacking classifier made of RF, NN, and bagging, which had a 97.4% accuracy rate. Moreover, PCA was applied to reduce the features before classification. It can be seen from Table 4 that [9] obtained an accuracy of 97.4% and F1 score of 97% while the proposed work is able to attain an accuracy of 97.07% and F1 score of 96.67%. The better performance of the model reported in [9] may be due to the use of a stacking classifier along with PCA.

The classifiers in this work are trained using an imbalanced dataset. However, the accuracy of such models can be deceptive. To overcome this issue, the dataset used in this research can be improved using sampling techniques.

Table 4: Comparison with previous works

| **Method / Evaluation Metrics** | [3] | [9] | **Proposed (XGBoost-Random Forest)** |
|---|---|---|---|
| Accuracy | 0.9643 | 0.974 | 0.9707 |
| F1 score | 0.9643 | 0.97 | 0.9667 |

# 7 Conclusion

Phishing attacks are the most frequent attacks that impact organizations globally. Many potential consequences may result from a successful phishing attempt. As a result of the breach, clients may leave the business causing huge financial losses. If an individual is targeted, their identity, credit, or financial information may be seriously compromised. Uprooting this cybercrime is an uphill task. It is vital to develop and implement contingency measures to protect against such breaches promptly. Users can be protected from malicious cyberattacks by identification and management of phishing websites. Consequently, this research analyzes Machine Learning-based techniques for detecting phishing websites. Eight classification algorithms comprising Naive Bayes, logistic regression, SVM, K-Nearest Neighbours, Decision Tree, AdaBoost, Random Forest, and XGBoost as well as the three hybrid models namely, RF-AdaBoost, XGBoost-AdaBoost, and XGBoost-RF are implemented and compared.

We evaluated the work using an imbalanced dataset, with a higher percentage of legitimate URLs than phishing URLs. This allowed us to evaluate the security offered in a realistic setting because of the higher probability of legitimate URLs in a practical scenario. The dataset is used to train and test each classification model and hybrid model. The classification technique using XGBoost-Random Forest yields the best accuracy, with a score of 97.07%. It can therefore be used to identify phishing websites more precisely. The development of ensemble models utilizing various robust machine learning algorithms and deep learning techniques will be the focus of future research.

## Authors' Information

– **Mukta Mithra Raj** is B.E. in Computer Science from the Birla Institute of Technology and Science Pilani, Dubai Campus, United Arab Emirates.

– **J. Angel Arul Jothi** is Assistant Professor at the Department of Computer Science of the Birla Institute of Technology and Science Pilani, Dubai Campus, United Arab Emirates.

[2] https://www.kaggle.com/datasets/akashkr/phishing-website-dataset

## Authors' Contributions

- **Mukta Mithra Raj** participated in devising the concept of the article and in analyzing and interpreting the data for the research.
- **J. Angel Arul Jothi** participated in revising critically the article and approved the final version to be published.

## Competing Interests

The authors declare that they have no competing interests.

## Funding

No funding was received for this project.

## References

[1] J. Fruhlinger, "What is phishing? examples, types, and techniques." https://www.csoonline.com/article/2117843/what-is-phishing-examples-types-and-techniques.html, 2022.

[2] IBM, "Cost of a data breach report 2021." https://www.dataendure.com/wp-content/uploads/2021_Cost_of_a_Data_Breach_-2.pdf, 2021.

[3] J. Gu and H. Xu, "An ensemble method for phishing websites detection based on xgboost," in *2022 14th International Conference on Computer Research and Development (ICCRD)*, pp. 214–219, IEEE, 2022. https://doi.org/10.1109/ICCRD54409.2022.9730579.

[4] A. Maini, N. Kakwani, B. Ranjitha, M. Shreya, and R. Bharathi, "Improving the performance of semantic-based phishing detection system through ensemble learning method," in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pp. 463–469, IEEE, 2021. https://doi.org/10.1109/MysuruCon52639.2021.9641614.

[5] A. Pandey, N. Gill, K. Sai Prasad Nadendla, and I. S. Thaseen, "Identification of phishing attack in websites using random forest-svm hybrid model," in *International conference on intelligent systems design and applications*, pp. 120–128, Springer, 2018. https://doi.org/10.1007/978-3-030-16660-1_12.

[6] A. Ramana, K. L. Rao, and R. S. Rao, "Stop-phish: an intelligent phishing detection method using feature selection ensemble," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 1–9, 2021. https://doi.org/10.1007/s13278-021-00829-w.

[7] H. Abusaimeh and Y. Alshareef, "Detecting the phishing website with the highest accuracy," *TEM Journal*, vol. 10, pp. 947–953, 2021. https://www.ceeol.com/search/article-detail?id=955454.

[8] N. Tabassum, F. F. Neha, M. S. Hossain, and H. S. Narman, "A hybrid machine learning based phishing website detection technique through dimensionality reduction," in *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–6, IEEE, 2021. https://doi.org/10.1109/BlackSeaCom52164.2021.9527806.

[9] A. Lakshmanarao, P. S. P. Rao, and M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 1164–1169, IEEE, 2021. https://doi.org/10.1109/ICAIS50930.2021.9395810.

[10] A. Subasi and E. Kremic, "Comparison of adaboost with multiboosting for phishing website detection," *Procedia Computer Science*, vol. 168, pp. 272–278, 2020. https://doi.org/10.1016/j.procs.2020.02.251.

[11] A. Zamir, H. U. Khan, T. Iqbal, N. Yousaf, F. Aslam, A. Anjum, and M. Hamdani, "Phishing web site detection using diverse machine learning algorithms," *The Electronic Library*, vol. 38, no. 1, pp. 65–80, 2020. https://doi.org/10.1108/EL-05-2019-0118.

[12] L. R. Kalabarige, R. S. Rao, A. Abraham, and L. A. Gabralla, "Multilayer stacked ensemble learning model to detect phishing websites," *IEEE Access*, vol. 10, pp. 79543–79552, 2022. https://doi.org/10.1109/ACCESS.2022.3194672.

[13] A. Makkar, N. Kumar, L. Sama, S. Mishra, and Y. Samdani, "An intelligent phishing detection scheme using machine learning," in *Proceedings of the Sixth International Conference on Mathematics and Computing*, pp. 151–165, Springer, 2021. https://doi.org/10.1007/978-981-15-8061-1_13.

[14] 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, 2021. https://doi.org/10.3390/s21248281.

[15] M. Al-Sarem, F. Saeed, Z. G. Al-Mekhlafi, B. A. Mohammed, T. Al-Hadhrami, M. T. Alshammari, A. Alreshidi, and T. S. Alshammari, "An optimized stacking ensemble model for phishing websites detection," *Electronics*, vol. 10, no. 11, p. 1285, 2021. https://doi.org/10.3390/electronics10111285.

[16] P. L. Indrasiri, M. N. Halgamuge, and A. Mohammad, "Robust ensemble machine learning model for filtering phishing urls: Expandable random gradient stacked voting classifier (ergsvc)," *IEEE Access*, vol. 9, pp. 150142–150161, 2021. https://doi.org/10.1109/ACCESS.2021.3124628.

[17] M. Alsaedi, F. A. Ghaleb, F. Saeed, J. Ahmad, and M. Alasli, "Cyber threat intelligence-based malicious url detection model using ensemble learning," *Sensors*, vol. 22, no. 9, p. 3373, 2022. https://doi.org/10.3390/s22093373.

[18] M. Korkmaz, O. K. Sahingoz, and B. Diri, "Detection of phishing websites by using machine learning-based url analysis," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, IEEE, 2020. https://doi.org/10.1109/ICCCNT49239.2020.9225561.

[19] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in *International Conference on Innovative Computing and Communications*, pp. 253–260, Springer, 2019. https://doi.org/10.1007/978-981-13-2354-6_27.

[20] R. R. Popat and J. Chaudhary, "A survey on credit card fraud detection using machine learning," in *2018 2nd international conference on trends in electronics and informatics (ICOEI)*, pp. 1120–1125, IEEE, 2018. https://doi.org/10.1109/ICOEI.2018.8553963.

[21] E. Bujokas, "Feature importance in decision trees." https://towardsdatascience.com/feature-importance-in-decision-trees-e9450120b445, 2022.