Komputika: Jurnal Sistem Komputer
Volume 13, Nomor 2, Oktober 2024, hlm. 233 - 243
DOI: 10.34010/komputika.v13i2.12495

p-ISSN: 2252-9039
e-ISSN: 2655-3198

Tree-based Ensemble Machine Learning for Phishing Website Detection

Husni Fadhilah[1*], Diky Restu Maulana[2], Rahayu Utari[3]

[1,2,3)] Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung, Indonesia 40132

**email: 23523034@std.stei.itb.ac.id*

(Naskah masuk: *11 Maret 2024*; direvisi: *12 Oktober 2024*; diterima untuk diterbitkan: *26 Oktober 2024*)

ABSTRACT *– Phishing websites are a major cybersecurity threat, deceiving users into revealing sensitive information by imitating legitimate websites. As these attacks evolve, accurate and efficient detection methods are crucial for mitigating their impact. Detecting phishing websites using traditional methods has limitations due to the increasing sophistication of phishing techniques. This research addresses the need for more advanced machine learning approaches to improve detection accuracy. The goal of this research is to explore the effectiveness of feature selection techniques and tree-based ensemble machine learning models, specifically Random Forest and Extra Trees, in identifying phishing websites. A comprehensive evaluation of the Random Forest and Extra Trees models was conducted. Feature selection techniques were applied to optimize the detection process, reducing complexity while maintaining high accuracy. The models were trained and tested on a dataset of phishing and legitimate websites, with performance metrics such as precision, recall, and accuracy being analyzed. The findings demonstrate that the Random Forest and Extra Trees models achieved an accuracy of 98.2%, outperforming other machine learning approaches in detecting phishing websites. These results suggest that feature selection, combined with ensemble learning models, can significantly enhance phishing website detection.*

Keywords *- Phishing website, ensemble tree, machine learning, feature selection, classification*

1. INTRODUCTION

In many nations, uninhibited entry to information outlets and public networks is considered a fundamental entitlement. The rise of the digital age has instigated substantial shifts in human behaviors, requiring adjustment. With the proliferation of e-commerce and online consumerism, phishing represents a significant cyber threat. The protection of our assets is anticipated to evolve into an essential element of contemporary civilization [1]. When users enter confidential information into imitation websites mirroring authentic ones, they inadvertently provide fraudulent entities access to their sensitive data, including credit card details, passwords, and other private information [2].

Phishing remains a prevalent and perilous form of cyber-attack, posing substantial dangers in the contemporary digital landscape. With the increasing dependence on online platforms for various endeavors such as business operations, transactions, and healthcare services, susceptibility to phishing attacks has risen [3]. These attacks involve the deceptive acquisition of personal and sensitive data, utilizing a combination of technical manipulation and social engineering strategies.

Phishing refers to a deceitful technique employed by attackers to acquire confidential information from unsuspecting individuals. Typically, phishing attacks leverage fraudulent emails or text messages, seemingly from trusted sources, deceiving unsuspecting individuals into divulging their confidential data [4]. These increasingly prevalent phishing websites mimic legitimate sites in appearance but are engineered to illicitly gather sensitive data provided by victims.

The alarming surge in phishing incidents over recent years underscores the urgent need for enhanced cybersecurity measures. Statistics from the Anti-Phishing Working Group (APWG) reveal a staggering rise in phishing attacks, with a notable tripling observed in 2022 compared to 2020. In the second quarter of 2023, the Anti-Phishing Working Group (APWG) observed 1,286,208 phishing attacks,

marking a decrease from 1,624,144 attacks in the previous quarter, which was the highest recorded. Despite being the third-highest quarterly total ever recorded, there was a significant decline by the end of the second quarter, with 306,847 attacks in June 2023, the lowest since November 2021 [5].

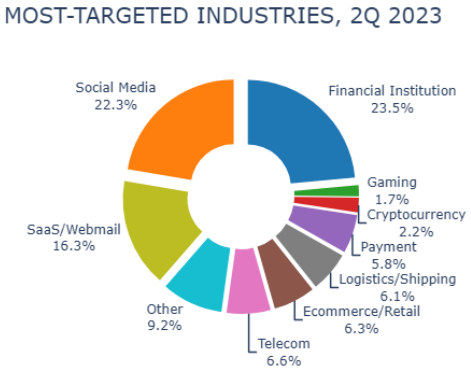

Figure 1 Statistics of phishing attacks and most-targeted industries, 2Q 2023 [5]

Recognizing the gravity of the situation, the machine learning community has turned its attention to the identification and classification of phishing websites. Through the development of advanced models and innovative approaches, researchers aim to bolster defenses against these malicious activities [6]. Techniques such as rule-based algorithms, heuristic analysis [4], URL-based methods, and machine learning-based solutions offer promising avenues to shield users from phishing attempts.

A comprehensive URL composition is essential to grasp how cyber attackers construct phishing domains. Uniform Resource Locators (URLs) are utilized to address web pages [7]. The standard structure of a URL is depicted in Table 1 [8]. By leveraging these diverse strategies, organizations can fortify their cybersecurity posture and safeguard sensitive information from falling into the wrong hands.

Table 1 Breaking down the entire URL string into smaller substrings

https://	example.com/	examples/	index.php	?q=example&y=2024
Protocol	**Domain**	**Directory**	**File**	**Parameters**

To combat the persistent menace of phishing, robust cybersecurity measures are indispensable, with artificial intelligence (AI) emerging as a promising avenue. Within AI, machine learning (ML) algorithms present an opportunity to identify and classify phishing attacks by scrutinizing historical data for patterns and markers of fraudulent behavior [6]. The utilization of ML models facilitates the augmentation of detection capabilities, enabling accurate discrimination between phishing websites and legitimate ones.

For practical implementations, it is crucial that these algorithms can accurately differentiate between phishing and authentic websites. Classification algorithms are frequently employed to categorize data into discrete classes or categories based on their inherent differences or similarities [4]. Clearly, among the multitude of machine learning techniques available, classification algorithms are particularly adept at tackling issues related to phishing. In the realm of phishing website detection, pertinent features are extracted from a variety of sources including URLs, DNS records, payload size, and WHOIS queries regarding domain names and IPs [4] [6] [9].

Many researchers use several strategies to detect malicious URLs using machine learning [10]. Swapna et al [11] employ various machine learning algorithms to identify key URL classification features. With a dataset of multiple URL records and 112 features, the research proposes a Machine Learning-based approach for feature selection. Using ensemble algorithms like Bootstrap Aggregation, Boosting, and stacking, the investigation compares and selects optimal features across five sections: preprocessing, dataset analysis, feature selection, algorithm comparison, and model evaluation. Achieving 93% accuracy, the model identifies 29 core features crucial for URL analysis.

Shaukat et al [12] present an effective layered classification model for website detection, utilizing URL structure, text, and image features. With a dataset of 20,000 website URLs, 22 key features are extracted from each URL, and a text evaluation dataset is prepared using NLP techniques to address the challenge of phishing websites embedding text as images. Experimental results show efficient phishing detection using XGBoost, achieving 94% accuracy and precision during training and 91% accuracy in testing. These results underscore the efficacy of the model in discerning between phishing and authentic websites, thereby making a substantial contribution to the early detection of complex phishing attacks and bolstering the security of internet users.

Wei et al [13] evaluate various classification models, including Random Forest (RF), on datasets with different feature sets. Results show RF outperforms deep learning models, achieving the highest testing accuracy rate of 96.94%. Additionally, RF demonstrates the lowest training time and minimal accuracy deterioration when using selected features. These findings validate RF's effectiveness in distinguishing between legitimate and phishing websites and underscore the potential of ensemble ML methods in real-time phishing detection.

Suleiman et al [14] introduce a deep learning-based approach for enhanced phishing detection, utilizing convolutional neural networks (CNNs) to

classify phishing sites with high accuracy. Assessment using a dataset consisting of 6,157 legitimate and 4,898 phishing websites illustrates the CNN models' effectiveness in identifying unrecognized phishing sites, surpassing conventional machine learning classifiers with a detection rate of 98.2% and an F1-score of 0.976. The suggested approach exhibits favorable comparisons to contemporary deep learning-based methods for phishing detection.

Table 2 Overview of prior studies leveraging the Phishing Dataset Mendeley 2020 (Acc = Accuracy, P = Precision, R = Recall, F1 = F1 Score)

#	Model	Acc (%)	P (%)	R (%)	F1 (%)
[11]	XGBoost	93	-	-	-
[12]	XGBoost	91	91	92.74	91.66
[13]	Random Forest	96.94	95.55	95.55	-
[7]	Light Gradient Boosting	97.83	96.81	98.93	97.86
[10]	Temporal Convolutional Network	98	97	97	97

This research contributes to the field of phishing website detection by proposing an analysis that focuses on feature selection techniques to enhance the model's performance and accuracy. The primary reason for focusing on feature selection techniques in this research stems from the critical role these techniques play in reducing the dimensionality of datasets, improving model interpretability, and enhancing computational efficiency without compromising accuracy [15, 16]. Feature selection is essential when dealing with phishing detection tasks, as it helps eliminate redundant or irrelevant features that could mislead the models and increase computational complexity. By focusing on the most informative features, this research ensures that the models are trained more effectively and can generalize better to unseen data [16]. This emphasis aligns with previous research that has highlighted the necessity of feature selection in phishing detection [2].

As for the choice of the Random Forest and Extra Trees models, these ensemble-based methods are known for their robustness, ability to handle large datasets, and capacity to work well with feature selection techniques [13, 15]. Random Forest is widely recognized for its capacity to handle high-dimensional data and perform well in classification tasks with minimal hyperparameter tuning [13]. Extra Trees, on the other hand, builds multiple de-correlated trees with randomized splitting criteria, which can result in more diversified predictions and potentially improve performance in complex datasets like those for phishing detection [15, 16]. Both models have shown strong performance in prior studies and are particularly effective in phishing website detection tasks, where the challenge lies in identifying subtle patterns that distinguish phishing URLs from legitimate ones [15].

In comparison to other machine learning models, these tree-based ensemble models excel in terms of accuracy and interpretability. Their performance has been proven superior in previous research, demonstrating higher detection rates and lower false positives in various phishing detection tasks [13]. Furthermore, they are computationally less expensive compared to deep learning models, making them more suitable for real-time phishing detection, which is a key concern in cybersecurity [13].

Compared to previous studies, this research builds upon well-established findings by integrating multiple feature selection techniques alongside ensemble-based machine learning models. Prior research has predominantly focused on model improvements without fully exploring the potential gains from integrating optimized feature selection processes [15, 16]. This research bridges that gap by not only optimizing models such as Random Forest and Extra Trees but also introducing new methods for refining the feature selection process, ensuring better accuracy and lower computational costs.

Moreover, the research explores the application of sampling techniques to address imbalanced datasets commonly encountered in phishing detection tasks. Additionally, this research investigates the optimization of hyperparameters for ensemble tree models, such as Random Forest and Extra Trees, to improve overall model performance. Through these contributions, the research aims to advance the effectiveness of phishing website detection methods, ultimately enhancing cybersecurity measures.

2. METHODOLOGY

In this research, a comprehensive and systematic methodology is employed to achieve the objectives aimed at enhancing phishing website detection using Tree-based Ensemble Machine Learning techniques. The focus is on distinguishing between two classes of websites: legitimate and phishing. Legitimate websites refer to authentic and trustworthy online platforms, whereas phishing websites are fraudulent entities designed to deceive users into divulging sensitive information. By accurately classifying websites into these two classes, robust detection models are developed to identify potential threats and safeguard users from cyber-attacks. Additionally, the efficiency of the methodology is assessed by comparing the outcomes with those of previous studies. These results provide valuable insights into the effectiveness and performance of

ensemble machine learning algorithms in detecting phishing websites, thereby contributing to the advancement of cybersecurity measures.

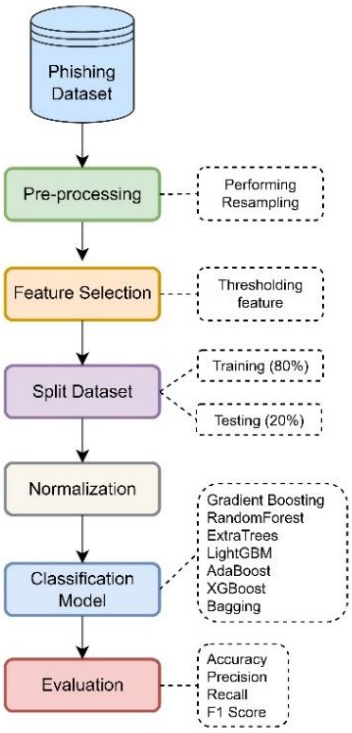

Figure 2 The steps of this research

The primary framework of this investigation is illustrated in Figure 2, comprising six distinct stages: pre-processing, feature selection, dataset splitting, normalization, model classification, and evaluation. The phishing dataset from Mendeley 2020 is utilized to develop and assess a range of classification methods aimed at detecting phishing websites. In the process of enhancing phishing website detection, several critical steps are undertaken. Initially, the dataset undergoes preprocessing, where resampling techniques are applied to address class imbalance issues commonly encountered in phishing datasets. Following this, methods for selecting features, specifically thresholding, are employed to detect and preserve the most pertinent features crucial for classification. Subsequently, the dataset is partitioned into training (80%) and testing (20%) subsets to streamline the processes of model training and evaluation. Normalization procedures are then applied to ensure uniformity and optimal performance across features. The dataset is then fed into a diverse array of classification models, including Gradient Boosting, RandomForest, ExtraTrees, LightGBM, AdaBoost, XGBoost, and Bagging, leveraging the strengths of ensemble learning to enhance detection accuracy. Finally, comprehensive evaluation metrics such as Accuracy, Precision, Recall, and F1 Score are employed to assess the performance of the models and ensure robust detection capabilities. Through this systematic approach, the detection of phishing websites is significantly bolstered, contributing to improved cybersecurity measures and safeguarding against online threats.

Data Collection

The dataset used in this research was meticulously collected from PhishTank and Alexa ranking to serve as a foundation for developing and assessing various classification methodologies tailored to identifying phishing websites [8]. It covers an extensive array of characteristics sourced from various aspects of uniform resource locator (URL) features, URL resolution metrics, and external services. Overall, the dataset encompasses 111 attributes, excluding the target phishing attribute, which indicates whether a specific instance is legitimate (0) or phishing (1). With a total of 88,647 instances, the dataset is balanced with 30,647 instances labeled as phishing and 58,000 instances labeled as legitimate. This balanced distribution reflects real-world scenarios where legitimate websites outnumber phishing ones, facilitating a more realistic evaluation of detection algorithms.

Table 3 Group of features present in the Phishing Dataset Mendeley 2020

#	No	Attribute Format	Desc
1	1-17	each number of ".-_/?=@&!~,+*#"$%" signs in the whole URL	Numeric
2	18-34	each number of ".-_/?=@&!~,+*#"$%" in domain	Numeric
3	35-51	each number of ".-_/?=@&!~,+*#"$%" in directory	Numeric
4	52-68	each number of ".-_/?=@&!~,+*#"$%" in file	Numeric
5	69-85	each number of ".-_/?=@&!~,+*#"$%" in parameters	Numeric
6	86–96	qty_vowels_domain, params_length, time_response, asn_ip, time_domain_activation, time_domain_expiration, qty_ip_resolved, qty_nameservers, qty_mx_servers, ttl_hostname,_qty_redirects	Numeric
7	97–102	qty_tld_url, number_of_characters_in_the_whole URL, domain_length, directory_length, file_length, params_length	Numeric
8	103 – 111	email_in_url, domain_in_ip, server_client_domain, tld_present_params, domain_spf, tls_ssl_certificate,	Boolean

url_google_index, domain_google_index, url_shortened

Data Pre-processing

Prior to model training, thorough data pre-processing is conducted to address various issues such as missing values, outliers, categorical variables, and duplicate values. These techniques ensure that the dataset is clean, standardized, and suitable for subsequent analysis. Additionally, feature engineering is performed to extract meaningful information from the raw data and derive new features that enhance the discriminative power of the models. Oversampling methods, such as Random Over Sampling, are employed to address the imbalance in the dataset, as depicted in Figure 3.

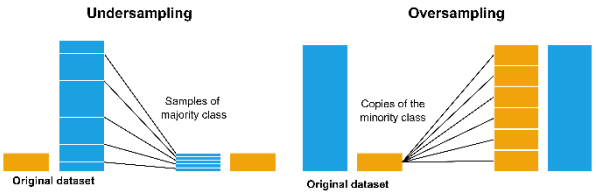

Figure 3 Illustration of resampling technique

Feature Selection

A key focus of the methodology is on feature selection, as identifying the most informative features is crucial for improving model performance and reducing computational complexity. A range of feature selection techniques is explored, including wrapper methods, filter methods, and ensemble-based approaches. These techniques allow the identification and retention of the most relevant features while discarding redundant or irrelevant ones, thus improving the efficiency and effectiveness of the models.

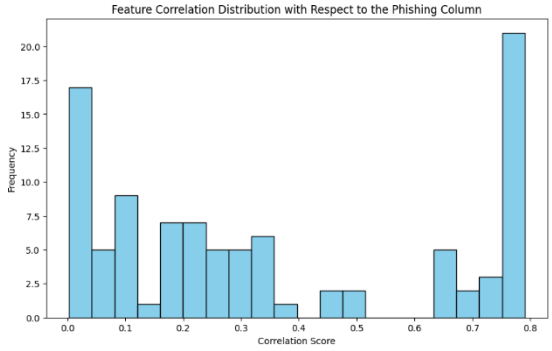

Figure 4 Feature correlation distribution concerning the phishing column

In this research, an approach is proposed where multiple correlation thresholds are tested to identify the optimal threshold for selecting the most informative features.

Algorithmic Feature Selection Process:

Let D represent the dataset containing n features F_1, F_2, ..., F_n, and a target label L (phishing column). The feature selection process involves testing a range of correlation thresholds T to determine the optimal value.

1. **Input**: Dataset D with n features and target label L.
2. **Output**: A subset $F_{selected}$ of relevant features and the best threshold T^*.
3. **Algorithm**:
   - Step 1: Initialize a set of candidate thresholds $T = \{T_1, T_2, \ldots, T_k\}$.
   - Step 2: For each threshold $T_j \in T$, apply the following procedure:
     - Step 2.1: For each feature $F_i \in D$, compute the correlation $C(F_i, L)$ between feature F_i and the target label L.
     - Step 2.2: If $C(F_i, L) > T_j$, include F_i in the selected features set $F_{selected}$ (T_j).
     - Step 2.3: Train the classification model using the selected features $F_{selected}$ (T_j).
     - Step 2.4: Evaluate the model using performance metrics such as accuracy, precision, recall, and F1-score.
   - Step 3: Identify the threshold T^* that yields the best performance based on the evaluation metrics.
   - Step 4: Return the selected features $F_{selected}$ (T^*) and the optimal threshold T^*.

Various correlation thresholds $T \in$ {-0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5} were tested, and the models were evaluated based on accuracy, precision, recall, and F1-score. After conducting experiments across different thresholds, the optimal threshold T^* = 0 yielded the highest performance. This threshold allowed for the selection of 98 features from the original 111 features. The removed columns include: qty_and_domain, qty_asterisk_domain, qty_comma_domain, qty_dollar_domain, qty_equal_domain, qty_exclamation_domain, qty_hashtag_domain, qty_percent_domain, qty_plus_domain, qty_questionmark_domain, qty_slash_domain, qty_space_domain, dan qty_tilde_domain.

Normalization

Before applying modeling techniques, the data was standardized using StandardScaler. The objective of StandardScaler is to transform the distribution of values within each feature to have a mean of 0 and a standard deviation of 1. Through this normalization process, uniformity in feature scales is achieved, preventing any single feature from dominating model computations. This fosters more effective learning, particularly with algorithms sensitive to scale variations [17]. Moreover, StandardScaler aids in accelerating the convergence of optimization algorithms like gradient descent by

rendering the fault contour surface more symmetric and accessible. Employing StandardScaler enhances the overall efficacy and reliability of machine learning models. The following formula (Equation 1) was applied to transform the features for the ensemble-based tree machine learning models.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Where z is a standardized value, x is the original value of the feature, μ is the average of the features, and σ is the standard deviation of the feature.

Classification Model

In the pursuit of optimizing model performance, a meticulous approach was adopted, leveraging grid search with 5-fold cross-validation to determine the optimal hyperparameters for the model exhibiting the highest accuracy. This rigorous methodology involved exhaustively searching through a specified parameter grid, systematically varying each hyperparameter while evaluating model performance using cross-validation. By partitioning the dataset into five subsets and iteratively training the model on four of them while validating the remaining subset, robustness was ensured, and overfitting was minimized. Through this iterative process, the hyperparameters yielding the highest accuracy across the cross-validated folds were identified, enhancing model generalization and predictive capability.

Table 4 Example hyperparameter values of Random Forest Classifier

#	Hyperparameter	Values
1	bootstrap	[True, False]
2	max_depth	[10, 20, None]
3	max_features	['auto', 'sqrt']
4	min_samples_leaf	[1, 2, 4]
5	min_samples_split	[2, 5, 10]
6	n_estimators	[100, 200]

Some of the ensemble techniques used by the author include:

A. Bagging

Breiman [18] introduced the bagging algorithm, also known as bootstrap aggregating, as one of the earliest and simplest ensemble learning (EML) techniques. It is particularly effective for small training datasets. By employing bootstrap sampling with replacement, this method creates random subsets of data to train multiple models simultaneously. The individual output models generated from these bootstrap samples are combined using a majority vote. Bagging can enhance the accuracy of machine learning models and applies to regression and classification tasks. It helps reduce variance and improve model robustness, especially when using decision trees, making it suitable for weak models with high variance [19].

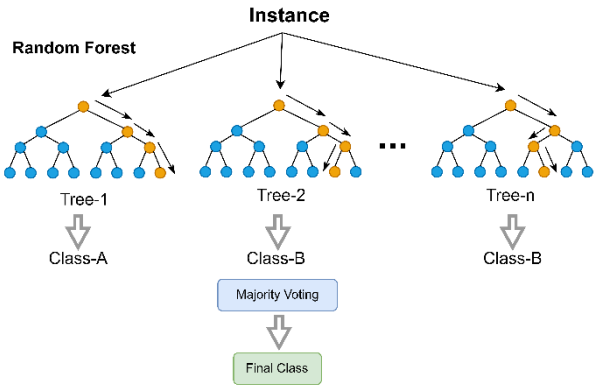

Figure 5 Example illustration of Random Forest

1. Random Forest

Random Forest is a technique in ensemble learning that leverages numerous decision trees to enhance predictive accuracy. Each decision tree is constructed using a random subset of both features and samples [20]. The collective prediction is determined by aggregating the predictions made by all trees.

The mathematical expression defining Random Forest is given by Equation 2:

$$\hat{y}_i = \frac{1}{B} \sum_{b=1}^{B} f_b(x_i) \quad (2)$$

where $\hat{y}_i$ is the prediction for sample i, B is the number of trees, f_b is the b-th decision tree, and x_i is the feature for sample i.

2. Extra Trees

Extremely Randomized Trees (also known as Extra-Trees (ET)) is a form of ensemble learning in machine learning, akin to Random Forest, albeit with certain distinctions. Much like Random Forest, Extra Trees harnesses numerous decision trees to enhance predictive accuracy. Nonetheless, in Extra Trees, each decision tree is crafted using a random subset of features and samples, and the node split at each junction is conducted randomly [19]. The mathematical expression governing Extra Trees remains identical to that of Random Forest.

B. Boosting

Boosting is a widely used ensemble learning technique aimed at improving the accuracy and performance of machine learning algorithms. It sequentially adds new models to the ensemble, effectively boosting weak learners into strong ones.

By mitigating overfitting in decision trees, Boosting reduces variance and bias while enhancing prediction accuracy. Like Bagging, Boosting involves creating multiple training datasets through random sampling with replacement from the original dataset. The sequence of models is then trained iteratively, with each new model adjusting the performance of the previous one. Finally, the weak learners are combined using weighted majority voting to form the final strong model. Notable examples of Boosting algorithms include AdaBoost, Categorical Boosting (CatBoost), Stochastic Gradient Boosting (SGB), Light Gradient Boosting (LightGBM), AdaBoost, and eXtreme Gradient Boosting (XGBoost) [19].

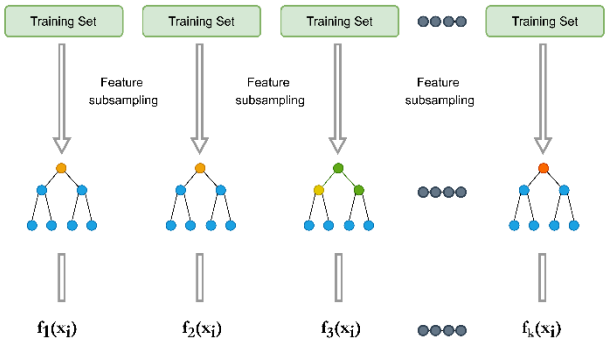

Figure 6 Example illustration of XGBoost

1. Gradient Boosting

Gradient Boosting is an ensemble learning technique that constructs predictive models by combining multiple relatively weak prediction models. These models typically consist of decision trees [21]. Gradient Boosting improves upon previous model errors by focusing on instances previously considered difficult to predict. The formula for Gradient Boosting is given by Equation 3:

$$\hat{y}_i = \sum_{k=1}^{K} f_{k(x_i)} \tag{3}$$

where $\hat{y}_i$ is the prediction for sample i, K is the number of models, f_k is the k-th model, and x_i is the feature for sample i.

2. LightGBM

LightGBM is a Gradient Boosting implementation made available by Microsoft under an open-source license [22]. Employing strategies like binning and leaf-wise growth, LightGBM enhances both training speed and predictive accuracy. The mathematical expression governing LightGBM mirrors that of Gradient Boosting [19].

3. AdaBoost

AdaBoost is an ensemble learning technique that was developed in 1995 by Freund and Schapire [23] is employs numerous weak models to enhance predictive accuracy. Each weak model is constructed on a random subset of features and samples, with weights assigned to each sample based on its difficulty level in prediction.

4. XGBoost

XGBoost is a Gradient Boosting implementation provided as an open-source tool by DMLC. Employing methods like pruning and cache-aware computing, XGBoost which was developed in 2016 by Chen and Guestrin [24] enhances both training speed and prediction accuracy. The mathematical expression governing XGBoost remains consistent with that of Gradient Boosting.

Evaluation and Validation

Finally, the trained models were evaluated using a range of performance metrics such as accuracy, precision, recall, and F1-score. Extensive validation experiments were conducted to assess the robustness and generalizability of the approach across different datasets and scenarios. By comparing the performance of the proposed approach with existing methods and benchmarks, its superiority and effectiveness in detecting phishing websites accurately and efficiently were demonstrated.

The determination of accuracy involves a straightforward calculation, typically achieved by dividing the sum of correct evaluations by the total number of evaluations. In essence, accuracy serves as a pivotal metric in assessing the performance of a model or system. To delve deeper into the intricacies of accuracy measurement, it becomes imperative to dissect the various components contributing to it, namely true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These elements form the fundamental pillars upon which accuracy is quantified, as delineated by the formulation depicted in Equation 4.

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{4}$$

Precision evaluates the accuracy of positive predictions by quantifying the proportion of true positive predictions relative to all positive predictions generated by the model. In essence, precision serves as a metric to gauge the reliability of positive predictions. Mathematically, precision is computed as the ratio of true positives (TP) to the total of true positives and false positives (FP), thereby providing a precise measure of the model's predictive performance in identifying true positive instances. Precision is computed as Equation 5:

$$Precision = \frac{TP}{(TP + FP)} \quad (5)$$

Recall, often referred to as sensitivity or true positive rate, signifies the ratio of true positive predictions to the total number of actual positive instances within the dataset. It characterizes the model's capability to correctly identify all pertinent cases within the dataset. Mathematically, recall is computed as the division of true positives (TP) by the sum of true positives and false negatives (FN), thereby providing insight into the model's effectiveness in capturing all relevant positive instances. Recall is calculated as Equation 6:

$$Recall = \frac{TP}{(TP + FN)} \quad (6)$$

The F1 score represents the harmonic mean of precision and recall, thereby offering a consolidated metric that strikes a balance between both precision and recall values. This composite measure is particularly valuable in scenarios where the dataset exhibits class imbalance, as it assigns equal significance to both precision and recall. By harmonizing these two metrics, the F1 score provides a comprehensive evaluation of the model's performance, accounting for both the precision of positive predictions and the ability to capture all relevant positive instances. The F1-score is calculated as Equation 7:

$$F1 = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \quad (7)$$

3. Result And Discussion

In this research, several ensemble-based tree classification algorithms were applied to the Mendeley Dataset after preprocessing and normalization steps. The model's performance and robust detection capabilities were evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics are instrumental in gauging the effectiveness of the models in distinguishing between phishing and legitimate websites. Table 5 shows the result comparison between different ensemble-based tree models.

Table 5 Performance metrics for different models (Acc = Accuracy, P = Precision, R = Recall, F1 = F1 Score)

Model	Train Acc (%)	Test Acc (%)	P (%)	R (%)	F1 (%)
Random Forest	99.99	98.22	98.23	98.22	98.22
Extra Trees	99.99	98.21	98.21	98.21	98.21
Bagging	99.87	98.05	98.05	98.05	98.05
XG Boost	98.10	97.38	97.38	97.38	97.38
Light GBM	97.12	96.79	96.79	96.79	96.79
Gradient Boosting	95.24	95.31	95.32	95.31	95.31
Ada Boost	93.39	93.57	93.59	93.57	93.57

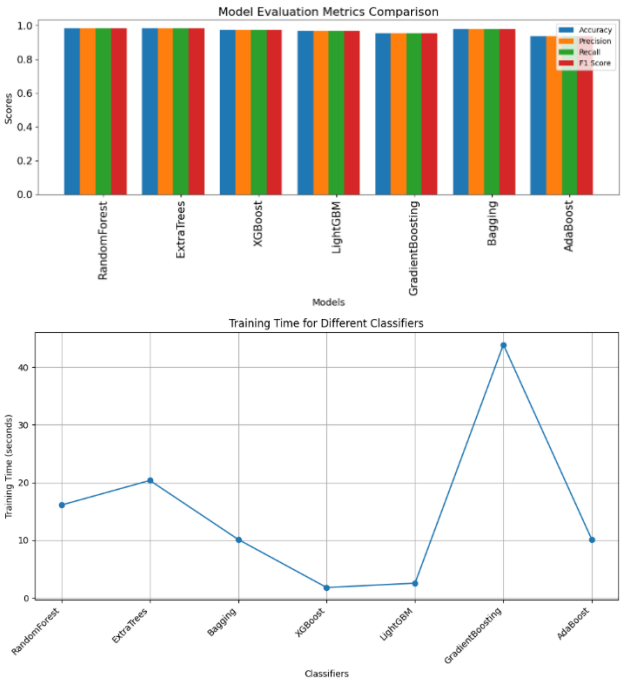

Figure 7 Metrics and training time comparison of different classifier

Random Forest and Extra Trees achieved the highest accuracy scores, both surpassing 98%. This indicates that these models were highly effective in correctly classifying instances. Moreover, they exhibited near-perfect precision scores, indicating a negligible number of false positive predictions. The recall and F1 scores for Random Forest and Extra Trees were also high, suggesting that they successfully identified a vast majority of positive instances while maintaining a balance between precision and recall. Despite their similar performance, Random Forest outperformed Extra Trees in terms of training time, completing the training process in approximately 11 seconds compared to Extra Trees in 20 seconds.

Bagging also demonstrated commendable performance with an accuracy score exceeding 98%. While its precision, recall, and F1 scores were slightly lower than Random Forest and Extra Trees, they still exhibited robust predictive capabilities. Bagging achieved a good balance between precision and recall, indicating its effectiveness in both minimizing false positives and capturing true positives. Similar to Random Forest, Bagging also showed efficient training time, completing the training process in approximately 11.5 seconds.

XGBoost and LightGBM, while exhibiting slightly lower accuracy scores compared to the

aforementioned models, still performed well with accuracy scores exceeding 97% and 96%. Although their precision, recall, and F1 scores were slightly lower than Random Forest, Extra Trees, and Bagging, they demonstrated competitive performance in capturing positive instances while maintaining low false positives. Notably, XGBoost exhibited significantly faster training time compared to other models, completing the training process in approximately 2 seconds, making it an attractive option for time-sensitive applications.

Gradient Boosting and AdaBoost achieved comparatively lower accuracy scores, indicating slightly inferior performance in comparison to other classifiers. Despite this, both models demonstrated respectable precision, recall, and F1 scores, indicating their ability to effectively capture positive instances while minimizing false positives. However, Gradient Boosting exhibited significantly longer training times compared to other models, with training time exceeding 45 seconds, which limits its suitability for real-time applications.

Based on the evaluation results using confusion matrices in Figure 8, it can be observed that the evaluated models exhibit varied performance in classifying the data. For instance, the RandomForest and ExtraTrees models have relatively low false negative (FN) counts, indicating that both models are quite adept at identifying true positive (TP) cases. On the other hand, the XGBoost, LightGBM, and Bagging models have relatively low false positive (FP) counts, suggesting that these three models tend to correctly classify a large number of true negative (TN) cases.

However, there are models such as GradientBoosting and AdaBoost that demonstrate lower performance compared to other models. The GradientBoosting model has relatively high FP and FN counts, indicating a tendency to misclassify both true positive and true negative cases. Similarly, the AdaBoost model also exhibits relatively high FP and FN counts, indicating issues in classifying both classes.

From these results, it can be inferred that the RandomForest and ExtraTrees models are more suitable for this classification task, given their relatively good performance in identifying both classes. However, further evaluation is needed to ensure the suitability of the models for specific application needs.

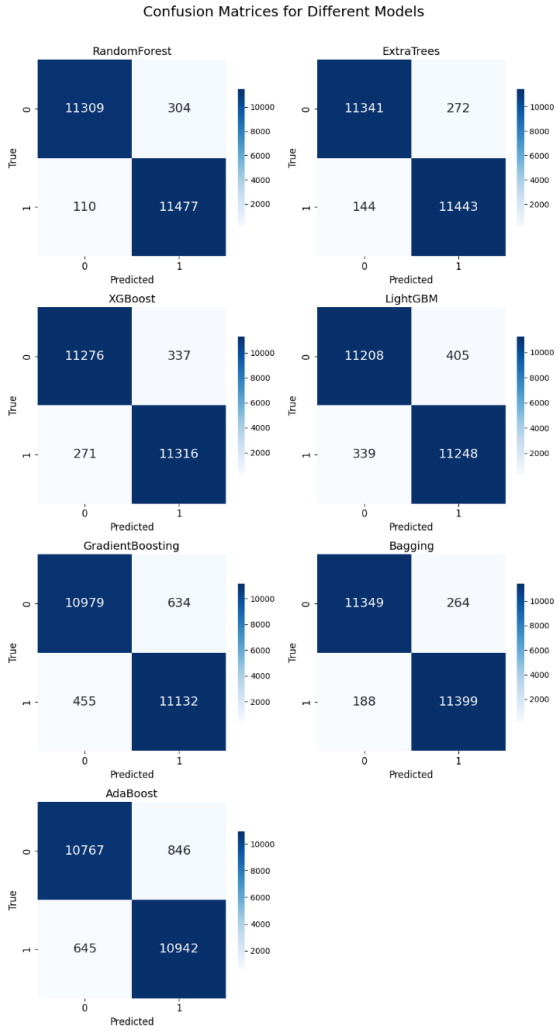

Figure 8 Confusion matrices for ensemble model

In summary, the results indicate that Random Forest, Extra Trees, Bagging, XGBoost, and LightGBM are promising classifiers for the task at hand, with Random Forest and Extra Trees exhibiting the highest accuracy and precision scores. XGBoost stands out for its fast training time, making it a favorable choice for applications requiring rapid model deployment. However, Gradient Boosting and AdaBoost, while achieving respectable performance, are less suitable for time-sensitive applications due to their longer training times. Overall, the findings provide valuable insights into the comparative performance of various classifiers and can inform the selection of an appropriate model based on specific requirements and constraints.

After examining the data and conducting analysis, it was found that the Extra Trees and Random Forest models exhibited excellent performance, with accuracy exceeding 98% and precision, recall, and F1 scores approaching or even reaching 98%. Additionally, Extreme Gradient Boosting (XGBoost) also demonstrated excellent performance with accuracy exceeding 97% and precision, recall, and F1 scores approaching 97%.

The best result was achieved by Random Forest and Extra Trees with the hyperparameter configuration: bootstrap = False, max_depth = 100, max_features = sqrt, min_samples_leaf = 2, min_samples_split = 5, and n_estimators = 400. When comparing these results with previous research, a significant performance improvement is observed in several cases. For instance, the XGBoost model developed achieved an accuracy of 97.38%, which is significantly higher compared to previous research that only achieved 91% [11] or 93% [12]. This provides confidence that the developed model has reached a state-of-the-art level in the classification task on the same phishing website detection dataset under research.

4. CONCLUSION

This research has contributed to the field of phishing website detection by proposing an approach based on tree-based ensemble machine learning techniques. The primary focus was on improving model performance and accuracy through the use of feature selection techniques, addressing imbalanced datasets, and optimizing hyperparameters for ensemble models such as Random Forest and Extra Trees. Notably, the Random Forest and Extra Trees models achieved an accuracy of 98.2%, showcasing their superior performance in detecting phishing websites. This improvement in detection accuracy reflects the robustness of the proposed feature selection methodology, which identified and retained only the most relevant features, ensuring efficient and accurate classification. These results align with the research objectives of enhancing model performance and developing reliable detection techniques, underscoring the potential of these models in accurately classifying phishing websites, distinguishing them from legitimate ones, and thereby strengthening cybersecurity measures to mitigate potential online threats.

Moving forward, several avenues for future research can be explored to further advance phishing website detection methods. Firstly, investigating the integration of advanced feature selection techniques and ensemble learning algorithms could enhance model performance and robustness. Additionally, exploring the utilization of deep learning approaches and neural networks offers new insights and opportunities for improved detection capabilities. Furthermore, incorporating real-time data sources and dynamic feature extraction methods enhances the adaptability of detection models to evolving phishing techniques.

REFERENCES

[1] S. Badotra and A. Sundas, "A systematic review on security of E-commerce systems," *Int. J. Appl. Sci. Eng.*, vol. 18, no. 2, pp. 1–19, 2021, doi: 10.6703/IJASE.202106_18(2).010.

[2] S. Shabudin, N. S. Sani, K. A. Z. Ariffin, and M. Aliff, "Feature selection for phishing website classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 587–595, 2020, doi: 10.14569/IJACSA.2020.0110477.

[3] K. Omari, "Comparative Study of Machine Learning Algorithms for Phishing Website Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 9, pp. 417–425, 2023, doi: 10.14569/IJACSA.2023.0140945.

[4] N. K.Subashini, "Phishing Website Detection Techniques: a Literature Survey," *J. Data Acquis. Process.*, vol. 38, no. 2, pp. 3329–3350, 2023, doi: 10.31862/9785426311961.

[5] APWG, "APWG Phishing Trends Report 2nd Quarter 2023," *Anti-Phishing Work. Gr.*, no. September, 2023.

[6] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 3, pp. 672–694, 2021, doi: 10.3390/make3030034.

[7] B. K. Gontla, P. Gundu, P. J. Uppalapati, K. Narasimharao, and S. M. Hussain, "A Machine Learning Approach to Identify Phishing Websites: A Comparative Study of Classification Models and Ensemble Learning Techniques," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 10, no. 5, pp. 1–9, 2023, doi: 10.4108/eetsis.vi.3300.

[8] G. Vrbančič, I. Fister, and V. Podgorelec, "Datasets for phishing websites detection," *Data Br.*, vol. 33, 2020, doi: 10.1016/j.dib.2020.106438.

[9] M. Bahaghighat, M. Ghasemi, and F. Ozen, "A high-accuracy phishing website detection method based on machine learning," *J. Inf. Secur. Appl.*, vol. 77, p. 103553, 2023, doi: 10.1016/j.jisa.2023.103553.

[10] M. A. Remmide, F. Boumahdi, N. Boustia, C. L. Feknous, and R. Della, "Detection of Phishing URLs Using Temporal Convolutional Network," *Procedia Comput. Sci.*, vol. 212, no. C, pp. 74–82, 2022, doi: 10.1016/j.procs.2022.10.209.

[11] N. S. Goud and A. Mathur, "Feature Engineering Framework to detect Phishing Websites using URL Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 295–303, 2021, doi: 10.14569/IJACSA.2021.0120733.

[12] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning," *Sensors*, vol. 23, no. 19, pp. 1–27, 2023, doi: 10.3390/s23198070.

[13] Y. Wei and Y. Sekiya, "Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection," *IEEE Access*, vol. 10, no. November, pp. 124103–124113, 2022, doi: 10.1109/ACCESS.2022.3224781.

[14] S. Y. Yerima and M. K. Alzaylaee, "High Accuracy Phishing Detection Based on Convolutional Neural Networks," *ICCAIS 2020 - 3rd Int. Conf. Comput. Appl. Inf. Secur.*, pp. 0–5, 2020, doi: 10.1109/ICCAIS48893.2020.9096869.

[15] Z. Farhadi, H. Bevrani, M. R. Feizi-Derakhshi, W. Kim, and M. F. Ijaz, "An Ensemble Framework to Improve the Accuracy of Prediction Using Clustered Random-Forest and Shrinkage Methods," *Appl. Sci.*, vol. 12, no. 20, 2022, doi: 10.3390/app122010608.

[16] P. Ponnusamy and P. Dhandayudam, "An Optimized Bagging Learning with Ensemble Feature Selection Method for URL Phishing Detection," *J. Electr. Eng. Technol.*, vol. 19, no. 3, pp. 1881–1889, 2024, doi: 10.1007/s42835-023-01680-z.

[17] J. Brownlee, "How to use StandardScaler and MinMaxScaler transforms in python," *Mach. Learn. Mastery*, 2020.

[18] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no., pp. 123–140, 1996, doi: 10.1007/BF00058655.

[19] M. Zounemat-Kermani, O. Batelaan, M. Fadaee, and R. Hinkelmann, "Ensemble machine learning paradigms in hydrology: A review," *J. Hydrol.*, vol. 598, no. December 2020, p. 126266, 2021, doi: 10.1016/j.jhydrol.2021.126266.

[20] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision tree and ensemble learning algorithms with their applications in bioinformatics," *Adv. Exp. Med. Biol.*, vol. 696, pp. 191–199, 2011, doi: 10.1007/978-1-4419-7046-6_19.

[21] J. Friedman, "Greedy Function Approximation : A Gradient Boosting Machine Author (s): Jerome H . Friedman Source : The Annals of Statistics , Vol . 29 , No . 5 (Oct ., 2001), pp . 1189-1232 Published by : Institute of Mathematical Statistics Stable URL : http://www," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[22] T.-Y. L. Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *31st Conf. Neural Inf. Process. Syst. (NIPS 2017), Long Beach, CA, USA*, 2017.

[23] R. E. Schapire, "A brief introduction to boosting," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2, no. 5, pp. 1401–1406, 1999.

[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.