TRANSMISI: JURNAL ILMIAH TEKNIK ELEKTRO, 26, (3), JULI 2024
p-ISSN 1411-0814 e-ISSN 2407-6422

ENHANCING BANK CUSTOMER PROTECTION AGAINST PHISHING ATTACKS THROUGH XGBOOST-BASED FEATURE ANALYSIS

Tan Regina Karin, Ramadhan Rahmat Sani[*)], Farrikh Al Zami dan Asih Rohmani

Informatics, Faculty of Informatics, Dian Nuswantoro University, Indonesia

[*]*Email: ramadhan_rs@dsn.dinus.ac.id*

Abstract

Internet usage in Indonesia has significantly increased, with approximately 175.4 million people or 64% of the population actively using the internet. While the internet provides numerous benefits, such as easy access to information and faster communication, this rise in usage also opens opportunities for cybercriminals to exploit user vulnerabilities. One of the most common forms of cybercrime is phishing, which attempts to steal users' personal information by impersonating a trusted entity. Current methods for detecting phishing are ineffective against zero-day phishing attacks. Therefore, this study employs the XGBoost algorithm to detect phishing websites. The dataset used consists of 30 features evaluated based on weight metrics, with only features scoring above or equal to 20 being selected. The research findings indicate that the XGBoost model, using feature selection techniques, can improve accuracy by 0.6% compared to using all features. Evaluation on selected features shows an accuracy of 95.5%, with precision, recall, and F1 scores all at 95.5%, 95.1%, and 95.3%, respectively. With these capabilities, XGBoost can be utilized to protect internet users from evolving phishing threats and assist financial institutions in anticipating customer losses.

Keywords: bank customers, feature selection, internet usage, phishing website, zero-day phishing attacks, XGBoost algorithm.

Abstrak

Penggunaan internet di Indonesia telah mengalami peningkatan signifikan, dengan sekitar 175,4 juta orang atau 64% dari populasi aktif menggunakan internet. Meskipun internet memberikan banyak manfaat seperti akses mudah ke informasi dan komunikasi yang cepat, kenaikan ini juga membuka peluang bagi penjahat siber untuk mengeksploitasi kerentanan pengguna. Salah satu bentuk kejahatan siber yang umum adalah phishing, di mana penyerang menyamar sebagai entitas yang terpercaya untuk mencuri informasi pribadi pengguna. Saat ini, metode deteksi phishing sering tidak efektif terhadap serangan zero-day. Oleh karena itu, penelitian ini menggunakan algoritma XGBoost untuk mendeteksi situs web phishing. Dataset yang digunakan terdiri dari 30 fitur yang dievaluasi berdasarkan metrik weight, dan hanya fitur-fitur dengan nilai di atas sama dengan 20 yang akan dipilih. Hasil penelitian menunjukkan bahwa model XGBoost, dengan teknik seleksi fitur, dapat meningkatkan akurasi 0,6 persen dari penggunaan semua fitur. Evaluasi pada fitur-fitur terpilih menunjukkan akurasi mencapai 95,5%, dengan presisi 95,5%, recall 95,1%, dan skor F1 95,3%. Dengan kemampuan ini, XGBoost dapat digunakan untuk melindungi pengguna internet dari ancaman phishing yang terus berkembang, serta membantu lembaga keuangan dalam mengantisipasi kerugian pelanggan.

Kata kunci: pelanggan bank, seleksi fitur, penggunaan internet, situs web phishing, serangan phishing zero-day, algoritma XGBoost

1. Introduction

In Indonesia, internet usage has seen a significant increase, with approximately 175.4 million people or 64% of the population actively using the internet [1]. This rise indicates that the internet has become an inseparable part of daily life for Indonesians, providing various benefits such as easy access to information, faster communication, and support for economic and educational activities. However, alongside these benefits, the internet also offers opportunities for cybercriminals to exploit user vulnerabilities. One of the most common forms of cybercrime is phishing [2]. Phishing is an attempt to steal users' personal information, such as passwords, credit card numbers, and other sensitive data, by impersonating a trusted entity [3]. This technique is often carried out through computer networks and the internet, targeting users via popular social networking sites like Facebook and Instagram, as well as through email. Phishers typically create fake websites that resemble legitimate ones to deceive users into entering their personal information [4]. While some internet users may be aware

of this threat, many still fail to recognize the signs of phishing, especially when the fake websites appear highly convincing. As a result, users who are less vigilant or lack adequate technical knowledge fall victim to phishing attacks.

In the past five years, since 2018, IDADX (Indonesia Anti-Phishing Data Exchange) has received reports of 106,806 phishing cases. One notable phishing incident targeted bank customers, resulting in their accounts being drained. Phishing cases cause significant losses to victims, as seen with 219 DBS Bank customers in Singapore who were defrauded in the first two weeks of this year, with total losses amounting to around S$446,000 or USD335,000. The victims received unsolicited SMS messages from both foreign and local numbers, with scammers posing as DBS or POSB Bank representatives warning of unauthorized account access attempts. The scammers sent links, urging customers to click to verify their identity and block suspicious transactions. Upon clicking the link, victims were directed to a fake DBS portal and asked to provide their internet banking details and One-Time Passwords (OTPs), which the scammers then used to drain their accounts. A similar incident occurred in Indonesia, specifically at BRI Branch Tabing in Padang City, in May 2022. A BRI Bank customer clicked a data collection link sent via WhatsApp, promising an exemption from a transaction fee of Rp. 150,000 per month with unlimited transactions. Consequently, without realizing it, the customer lost Rp 1.1 billion from their savings account. These cases highlight the seriousness of phishing threats and the financial losses they can cause. Scammers continuously develop techniques to deceive victims, such as sending messages from seemingly legitimate numbers and creating fake websites that closely resemble the originals.

The current problem in detecting phishing websites is the inability of list-based techniques and visual similarity to detect zero-day phishing attacks due to the short-lived nature of phishing websites. This indicates the need for more advanced detection techniques. One common method used is updating the blacklist of URLs or IP addresses in antivirus databases. However, attackers continually attempt to evade detection using creative techniques such as obfuscation and fast-flux. The weakness of this method lies in its inability to detect phishing attacks at an early stage (zero-hour phishing attack). Consequently, heuristic-based detection techniques have also been developed to capture some zero-hour phishing attacks by looking for common patterns in the attacks [5]. However, not all attacks display the same patterns, leading to a high rate of false positives. Therefore, security researchers are now focusing on developing machine learning techniques. Using this technology, algorithms can analyze past data and predict future attacks, including zero-hour phishing attacks. With this approach, phishing website detection can be done more accurately by analyzing blacklisted and legitimate URLs and their features [6].

Machine learning includes several techniques, one of them is classification. Classification is a commonly used method to group certain items based on similar characteristics. One standout classification method is eXtreme Gradient Boosting (XGBoost). XGBoost can handle various examples of classification, regression, and ranking with excellent results [7]. One of the main advantages of XGBoost is its robustness against outliers, which often pose a problem in other classification techniques. Additionally, XGBoost has shorter computation times and produces accurate predictions. The boosting method using XGBoost has been proven to provide better accuracy and processing time compared to other classification methods.

This study uses the XGBoost algorithm as the main technique for detecting phishing websites due to its proven advantages in various previous studies. In the research conducted by Jan Melvin Ayu Soraya Dachi and Pardomuan Sitompul titled "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," XGBoost achieved a perfect score (1.0) on all evaluation metrics for both 10,000 and 100,000-sized datasets, while Random Forest showed performance decline on small unbalanced datasets [8]. The study by Muhammad Kaddafi Nasution, Rd. Rohmat Saedudin, and Vandha Pradwiyasma Widartha in "Perbandingan Akurasi Algoritma Naïve Bayes dan Algoritma XGBoost pada Klasifikasi Penyakit Diabetes" showed that XGBoost had a higher accuracy, reaching 90.10%, compared to Naïve Bayes, which only reached 79.68% [9]. Liyana Mat Rani, Cik Feresa Mohd Foozy, and Siti Noor Baini Mustafa in the study titled "Feature Selection to Enhance Phishing Website Detection Based on URL Using Machine Learning Techniques" used TreeSHAP and Information Gain to rank features and select the top 10, 15, and 20 features. These features were then fed into three machine learning classifiers: Naïve Bayes, Random Forest, and XGBoost. They found that XGBoost achieved the highest detection accuracy of 98.59% using 15 URL features. [10]. The study by Ouedraogo Pengdwende Leonel Camille and Ganesh Gupta titled "URL Based Malicious Activity Detection Using Machine Learning" showed that XGBoost had the highest detection accuracy compared to models like Random Forest and Light GBM, making it highly effective in detecting malicious activities on URLs [11]. Additionally, the research by Y. Li & Chen in "A comparative performance assessment of ensemble learning for credit scoring" concluded that XGBoost and Random Forest are high-performance estimators in classification and regression, capable of preventing overfitting and handling missing and unbalanced data well [12]. The study by Sumitra Das Guptta and colleagues in "Modeling Hybrid Feature-Based Phishing

Websites Detection Using Machine Learning Techniques" revealed that the phishing detection approach using XGBoost was more effective, with a detection accuracy rate of 99.17%, far surpassing traditional approaches [13]. This evidence supports the selection of XGBoost in this study for phishing website detection due to its accuracy, speed, and ability to prevent overfitting. XGBoost becomes a highly effective tool in classifying whether a URL or website falls into the phishing category, considering its various features. Although XGBoost stands out with shorter computation times, its ability to produce highly accurate predictions is crucial in the effort to detect phishing sites.

The difference between this research and the previous one is the use of feature importance to emit and select the most relevant features in the process of detecting phishing websites based on URLs. Using feature selection techniques, this study aims to improve phishing detection efficiency by identifying the most influential features in website classification. Feature importance helps reduce model dimensions and makes the classifier more effective by eliminating less relevant attributes or those with minimal impact on the learning process [14]. Thus, only the most informative and relevant features are considered in the detection process, enhancing the overall accuracy and efficiency of the model. To rank feature importance, this study uses metrics such as weight, which represents the frequency of a feature's use in splitting data across all trees in the model. The weight metric provides a clear picture of each feature's contribution to the model's decision-making process. Features that are used more frequently indicate that they have a significant impact on data separation and help the model make more accurate predictions. Additionally, weight helps identify features that truly affect model performance, allowing less significant features to be eliminated without sacrificing accuracy, thereby increasing model efficiency and reducing computational complexity. Therefore, the use of feature importance is key in optimizing the performance of the XGBoost model for phishing detection, resulting in highly accurate predictions with shorter computation times. The combination of high speed and accuracy is crucial in protecting internet users from evolving threats and minimizing the adverse impacts of phishing attacks. This can provide significant support for banks in their efforts to anticipate customer losses and enhance security by distinguishing phishing websites from legitimate.

2. Research Methodology

In this study, a framework is needed to explain the research steps undertaken. Figure 1 shows the proposed research flow.

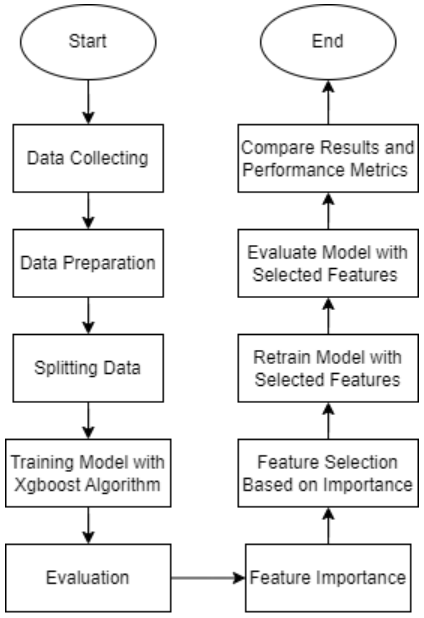

Figure 1. Research Flow

2.1. Data Collecting

The dataset used is the Phishing Websites dataset obtained from the UCI Machine Learning Repository. This dataset includes 11,055 records and consists of 30 different features, designed to help detect and classify phishing websites. The dataset was collected from various primary sources to ensure data accuracy and reliability. These sources include the Phishtank archive, a community database containing information on phishing websites, the MillerSmiles archive, which also maintains data on identified phishing websites, and Google search operators to find additional phishing websites not listed in the previous archives.

2.2. Data Preparation

Data preparation is a crucial step in this research method, aiming to ensure that the data used is clean, consistent, and ready for further analysis. The first step in data preparation is handling missing values. Missing values in the dataset can cause bias and reduce model performance. Therefore, it is essential to identify and properly handle missing values. This approach ensures that the dataset remains intact without losing significant information. The second step is removing duplicate data. Duplicate data can cause bias in the model and reduce prediction accuracy. To address this, rows with identical values in all columns are identified and removed. Thus, each record in the dataset is unique, which helps maintain data quality and improve model performance. The third step is removing features with constant values. Features with constant values do not provide useful information for the model because all values in these features are the same. Therefore, these features are identified and removed from

the dataset. Removing constant features improves model efficiency by reducing the number of features to be processed without sacrificing useful information.

2.3. Splitting Data

Splitting data is an important process in machine learning where the dataset is divided into subsets for training, validation, and testing the model. This division is crucial to objectively evaluate model performance and prevent overfitting. Without data splitting, the model tends to become too specific to the training data and may not perform well on new data. Data splitting is done before the model training process to ensure that the model can generalize well to unseen data. Generally, the dataset is divided into two main subsets: the training dataset and the testing dataset. The data splitting is performed in a stratified manner, with a certain proportion to maintain the same class distribution between the two subsets. In this study, the data is split with a proportion of 70% for training data and 30% for testing data. The 70-30 ratio has become a common practice in the machine learning community because it promotes the optimization of two crucial aspects in model development. Using 70% of the data for training ensures that the model gains a deep understanding of patterns and variations within the dataset, thereby enhancing its ability to make accurate predictions. Meanwhile, allocating 30% of the data for testing provides an opportunity to evaluate how well the model can generalize the learned information to new, unseen data. This approach not only supports statistical validity in assessing model performance but also reduces the risk of overfitting that may occur when the model becomes too specific to the training data.

2.4. XGBoost

XGBoost, or Extreme Gradient Boosting, is a highly effective and widely used technique in machine learning for prediction and classification, utilizing decision tree structures as its foundation. This algorithm is one of the boosting methods consisting of several interdependent decision trees, where each new tree built aims to correct the errors of the previous tree. This makes XGBoost particularly robust in handling data with high complexity and noise. The XGBoost model training process begins with initializing the model using predefined parameters. Some important parameters to consider include:

1. objective: Determines the type of task to be solved. In this case, 'binary:logistic' is used for binary classification.
2. eval_metric: Specifies the evaluation metric used. For instance, 'logloss' is used to measure log loss, which indicates how well the model performs in terms of prediction probabilities.
3. max_depth: The maximum depth of the decision trees. This parameter affects the model's complexity and its ability to capture information from the data.
4. eta (learning rate): Controls the step size when updating weights. This parameter is crucial to prevent overfitting by slowing down the learning process so that the model does not quickly overfit the training data.
5. n_estimators: The number of trees to be built in the boosting process. The more trees, the stronger the model, but it also increases the risk of overfitting.
6. subsample: The ratio of the training data sample used to build each tree. Subsampling helps reduce overfitting by ensuring each tree is built from different subsets of data.
7. colsample_bytree: The ratio of columns randomly selected when building each tree. This introduces diversity into the model, which also helps in reducing overfitting.

After initialization, the model is trained using the training data. This training process involves optimizing the objective function to minimize prediction errors (log loss) through boosting techniques. The objective function optimized in XGBoost is a combination of the loss function and a regularization term, which is expressed in equation (1):

$$Obj(\theta) = \sum_{i=1}^{n} l(yi, \hat{y}i) \sum_{k=1}^{K} \Omega(fk) \quad (1)$$

In each iteration, the gradient of the loss function is calculated for each sample, and a new model is added to reduce the residual error from the previous model [15]. This process can be summarized with the following formula (2):

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \quad (2)$$

This process continues until the desired number of trees is reached or until there is no significant improvement in the model's performance on the validation data. With this approach, XGBoost is capable of producing highly accurate and robust models, often outperforming other machine learning methods in various types of prediction and classification tasks.

2.5. Evaluation

After the model is trained, the next step is to conduct an initial evaluation to measure the model's performance using the test data. In this stage, predictions are made on the test data, and the results are compared with the actual values to assess the model's accuracy. Various evaluation metrics are used to get a comprehensive view of the model's performance, including accuracy, precision, recall, and F1-score

To visualize the model's prediction results, a confusion matrix is used. The confusion matrix helps in understanding the distribution of correct and incorrect

predictions made by the model, breaking down the results into four main categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). True positives indicate the number of correct positive predictions, false positives indicate the number of incorrect positive predictions, true negatives indicate the number of correct negative predictions, and false negatives indicate the number of incorrect negative predictions [16]. The confusion matrix can clearly identify where the model succeeds and where it may still need improvement. Here are the formulas for calculating some evaluation metrics:

1. Accuracy: Measures the proportion of total correct predictions (3).
2. Precision: Measures the proportion of correct positive predictions (4).
3. Recall: Measures the proportion of actual positives detected by the model (5).
4. F1-score: Combines precision and recall into a single metric using the harmonic mean (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$Accuracy = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

2.6. Feature Importance

The next step is to evaluate the importance of features to understand the contribution of each feature in making predictions. Features with low importance can be removed to simplify the model without significantly reducing performance. Using this method, we can identify which features contribute most to the model's predictions [17]. In this study, the metric used to evaluate the importance of features is weight. The weight metric indicates the frequency of using a feature to split nodes in all decision trees within the model. This provides a clear picture of how often a particular feature contributes to the model learning process.

After identifying less important features, the next step is to retrain the model using the remaining feature subset. Then, the retrained model is evaluated again to ensure that removing less important features does not significantly reduce model performance. Often, this process can actually improve model performance by reducing noise and overfitting. Using this approach, it can be ensured that the model works optimally with the most relevant features, thus providing accurate and efficient predictions. Effective feature evaluation and selection not only help simplify the model but also improve its overall performance.

3. Results and Discussion

The data collection process is a crucial initial step in machine learning, where relevant and high-quality data is gathered for further analysis. The dataset used in this study consists of 11,055 rows and 31 columns, where each column represents features used for classification, and one column 'result' serves as the target label. All data is in numeric format, facilitating the analysis and modeling process. After data collection, checks are performed to ensure data quality and integrity. From the checks conducted, it was found that there are no missing values in this dataset. Additionally, the dataset is also examined to detect duplicate data, which can affect model outcomes if not removed. It was found that there were 5,206 duplicate data entries in the dataset, which were then removed to ensure accurate analysis. Further examination is carried out to detect constant features, i.e., features that have the same value for all samples. Such features do not provide useful information for the model and should be removed. However, in this dataset, there is no constant features were found, so all existing features can be used in the modeling process. The class distribution in the target column 'result' is also checked to ensure that the data is not imbalanced. A balanced class distribution ensures that the model is not biased towards a particular class. In this dataset, the class distribution is as follows:

1. Class 1 (positive): 6,157 samples
2. Class 0 (negative): 4,898 samples

This fairly balanced distribution indicates that no additional handling for imbalanced data such as SMOTE (Synthetic Minority Over-sampling Technique) is needed. Balance in class distribution is crucial as it ensures that the model can learn well from both classes and does not provide biased predictions.

After removing duplicates from the initial dataset, we have 5,849 unique entries. Then the dataset is split into 2 parts: 30% for test data and 70% for training data. This means around 1,755 entries will be used for test data and around 4,094 entries will be used for training data. Test data will be used to evaluate the model's performance after training. The parameters used for the XGBoost model are as follows:

1. Objective: 'binary:logistic', indicating that the model will perform binary classification using logistic function as the objective function.
2. Eval_metric: 'logloss', used to measure the model's prediction quality. Logloss is a commonly used metric for classification model evaluation.
3. Max_depth: 4, which is the maximum depth of each tree in the model.
4. Eta: 0.1, which is the learning rate to control how fast the model learns from the data.
5. Seed: 42, used to reproduce the same results every time we run the model.

6. N_estimators: 100, which is the number of trees to be built in the model. This is the number of boosting rounds to be performed.
7. Subsample: 1, indicating that all samples will be used to train each tree.
8. Colsample_bytree: 1, indicating that all features will be used to train each tree.

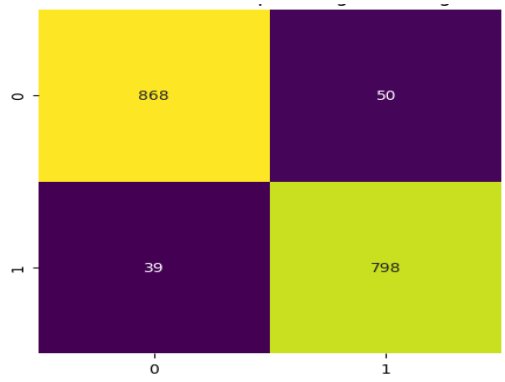

Figure 2. Confusion Matrix with All Features

Figure 2 depicts the visualization of the model prediction results using a confusion matrix. A confusion matrix is a table used to evaluate the performance of a classification model by comparing the model's predictions with the actual values from the test data. Based on the results of this confusion matrix, we can draw several important conclusions regarding the model's performance in classifying websites as legitimate or phishing.

1. True Negative (TN): There are 868 legitimate websites correctly identified by the model as legitimate websites. This indicates that the model has a good ability to recognize legitimate websites, minimizing the risk of legitimate websites being falsely flagged as phishing.
2. False Positive (FP): There are 50 legitimate websites incorrectly identified by the model as phishing websites. This indicates some errors where the model mistakenly considers safe websites as phishing.
3. False Negative (FN): There are 39 phishing websites incorrectly identified by the model as legitimate websites. This is a fairly dangerous situation as undetected phishing sites can endanger users by stealing their sensitive information.
4. True Positive (TP): There are 798 phishing websites correctly identified by the model as phishing sites. This indicates that the model is effective in detecting most phishing websites, helping protect users from potential threats and scams.

With a total of 1,755 tested data, the model's performance metrics are as follows:

1. Accuracy: 0.949 (94.9%)
2. Precision: 0.941 (94.1%)
3. Recall: 0.953 (95.3%)
4. F1 Score: 0.947 (94.7%)

The next step is to perform feature selection to improve the performance of the classification model. By using the feature importance technique with the weight metric, we can identify the most influential features in determining the classification of websites as legitimate or phishing. Feature importance ranks each feature based on its contribution to the model's predictions.

Based on the feature importance metric measurements, the ranking of features is as follows:

1. URL_of_Anchor: 112.0
2. web_traffic: 103.0
3. SSLfinal_State: 98.0
4. Prefix_Suffix: 78.0
5. Links_in_tags: 65.0
6. having_Sub_Domain: 64.0
7. Links_pointing_to_page: 63.0
8. SFH: 59.0
9. Request_URL: 40.0
10. having_IP_Address: 39.0
11. Page_Rank: 35.0
12. DNSRecord: 28.0
13. Google_Index: 27.0
14. URL_Length: 27.0
15. Domain_registration_length: 26.0
16. HTTPS_token: 24.0
17. age_of_domain: 22.0
18. popUpWidnow: 21.0
19. Submitting_to_email: 20.0
20. double_slash_redirecting: 19.0
21. Redirect: 18.0
22. Statistical_report: 15.0
23. Abnormal_URL: 15.0
24. Iframe: 13.0
25. having_At_Symbol: 13.0
26. on_mouseover: 13.0
27. Shortening_Service: 12.0
28. Favicon: 8.0
29. RightClick: 5.0
30. port: 2.0

After identifying the feature importance, the next step is to remove features with low importance values. In this case, the researcher decides to remove features with importance values below 20. This is done to simplify the model and reduce complexity without sacrificing prediction performance. The selected features are URL_of_Anchor, web_traffic, SSLfinal_State, Prefix_Suffix, Links_in_tags, having_Sub_Domain, Links_pointing_to_page, SFH, Request_URL, having_IP_Address, Page_Rank, DNSRecord, Google_Index, popUpWindow, age_of_domain URL_Length, Domain_registration_length, HTTPS_token, and Submitting to email. After removing features with low importance values, the remaining features are then reprocessed using the classification model, and the results are visualized through the following confusion matrix.

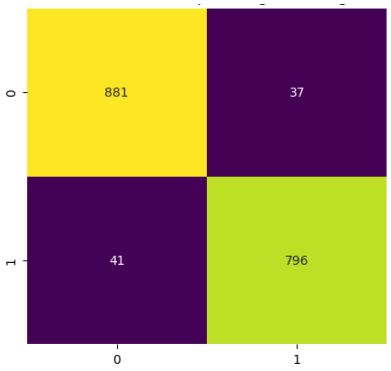

Figure 3. Confusion Matrix with Reduced Feature

Based on the confusion matrix results in figure 3, it can be concluded that:

1. True Negative (TN): There are 881 legitimate websites correctly identified by the model as legitimate. This indicates that the model has a good ability to recognize legitimate websites, minimizing the risk of legitimate websites being incorrectly flagged as phishing.
2. False Positive (FP): There are 37 legitimate websites incorrectly identified by the model as phishing websites. This indicates some errors where the model considers actually safe websites as phishing.
3. False Negative (FN): There are 41 phishing websites incorrectly identified by the model as legitimate websites. This is a potentially dangerous situation because undetected phishing sites can endanger users by stealing their sensitive information.
4. True Positive (TP): There are 796 phishing websites correctly identified by the model as phishing sites.

The final step is to compare the results from two sets of features, namely "All Features" and "Reduced Features". "All Features" represents all the features used in the model training process, while "Reduced Features" represents the result of training the model using selected features. From the obtained results in table 1, it is evident that the model with reduced features outperforms the model using all features. This is marked by an improvement in the values of all evaluation metrics. The accuracy value increased from 94.9% to 95.5%, the precision value increased from 94.1% to 95.5%, and although the recall value slightly decreased from 95.3% to 95.1%, the F1 Score value also increased from 94.7% to 95.3%. This improvement indicates that reducing features not only reduces model complexity but also enhances the overall model performance.

Table 1. Comparison of Metrics

Metric	All Features	Reduced Features
Accuracy	0.949288	0.955556
Precision	0.941038	0.955582
Recall	0.953405	0.951016
F1 Score	0.947181	0.953293

4. Conclusion

Based on the evaluation results, the developed model achieved an accuracy of 94.9% using all features, which increased to 95.5% after removing less important features. These results demonstrate that XGBoost is effective in classifying websites as legitimate or phishing. Using specific parameters in the XGBoost algorithm significantly influences the detection outcomes of phishing websites. Setting the objective to 'binary:logistic' enables the model to perform binary classification using logistic regression, which is effective in predicting the probability of phishing sites. Evaluating the model with the 'logloss' metric ensures prediction quality by measuring the accuracy of predicted class probabilities. Setting max_depth to 4 controls the complexity of decision trees to prevent overfitting, while eta set to 0.1 adjusts the learning rate to enhance the model's ability to generalize to new data. With n_estimators set to 100, the XGBoost model builds an adequate number of decision trees to capture complex patterns in the data. Using subsample and colsample_bytree both set to 1 ensures that the entire dataset and all features are optimally utilized during model training, supporting the model's ability to identify crucial patterns for phishing detection. Furthermore, feature selection techniques help simplify the model without sacrificing predictive performance, enabling the use of a more efficient and effective model in preventing financial losses for bank customers. By combining phishing website detection using the XGBoost model with prevention measures and education, banks can significantly reduce the risk of losses for their customers and enhance the security of their banking systems. Banks can proactively identify and block phishing sites before customers fall victim, thereby protecting them from potential identity theft and financial losses. XGBoost, with its ability to handle real-time data and continuous learning, can assist banks in detecting zero-hour attacks, i.e., attacks that are newly launched and not yet detected by previous systems. The use of multi-factor authentication systems can also be reinforced for high-risk transactions. These systems ensure that even if fraudsters manage to obtain customer login credentials, they still require additional confirmation to complete transactions, adding an extra layer of security. This comprehensive approach not only protects customers from phishing threats but also enhances their trust in digital banking services. Therefore, this research makes a significant contribution to improving banking system security through early detection and prevention of phishing attacks.

References

[1]. A. Susilo Yuda Irawan, N. Heryana, H. Siti Hopipah, D. Rahma Putri, and J. Hs Ronggo Waluyo Puseurjaya Telukjambe Timur Karawang Jawa Barat, "Identifikasi Website Phishing dengan Perbandingan Algoritma Klasifikasi," 2021. [Online]. Available: www.phishtank.com

[2]. M. Jonathan, S. Rostianingsih, and H. Novianus Palit, "Pengaruh Feature Selection terhadap Kinerja C5.0, XGBoost, dan Random Forest dalam Mengklasifikasikan Website Phishing," 2022.

[3]. N. B. Putri and A. W. Wijayanto, "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing," *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, pp. 59–66, Jan. 2022, doi: 10.34010/komputika.v11i1.4350.

[4]. R. Ester, S. Lina, and M. Sitio, "OPTIMASI ALGORITMA KLASIFIKASI DECISION TREE (CART) DENGAN METODE BAGGING UNTUK DETEKSI WEBSITE PHISHING," 2024. [Online]. Available: http://ojsamik.amikmitragama.ac.id

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

[6]. L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3. MDPI, pp. 672–694, Sep. 01, 2021. doi: 10.3390/make3030034.

[7]. A. P. Rosyadi, W. Maharani, and P. H. Gani, "PERSONALITY DETECTION ON TWITTER USER USING XGBOOST ALGORITHM," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 1, pp. 69–75, 2024, doi: 10.52436/1.jutif.2024.5.1.1166.

[8]. Jan Melvin Ayu Soraya Dachi and Pardomuan Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *JURNAL RISET RUMPUN MATEMATIKA DAN ILMU PENGETAHUAN ALAM*, vol. 2, no. 2, pp. 87–103, Oct. 2023, doi: 10.55606/jurrimipa.v2i2.1336.

[9]. Nasution M, Rohmat Saedudin R, and Widartha V, "PERBANDINGAN AKURASI ALGORITMA NAÏVE BAYES DAN ALGORITMA XGBOOST PADA KLASIFIKASI PENYAKIT DIABETES," vol. 8, no. 5, pp. 9765–9772, Oct. 2021.

[10]. L. Mat Rani, C. F. Mohd Foozy, and S. N. B. Mustafa, "Feature Selection to Enhance Phishing Website Detection Based On URL Using Machine Learning Techniques," *Journal of Soft Computing and Data Mining*, vol. 4, no. 1, pp. 30–41, May 2023, doi: 10.30880/jscdm.2023.04.01.003.

[11]. T. Z. Difaizi, O. P.-W. L. Camille, T. C. Benhura, and G. Gupta, "URL Based Malicious Activity Detection Using Machine Learning," in *2023 International Conference on Disruptive Technologies (ICDT)*, IEEE, May 2023, pp. 414–418. doi: 10.1109/ICDT57929.2023.10150899.

[12]. Y. Li and W. Chen, "A Comparative Performance Assessment of Ensemble Learning for Credit Scoring," *Mathematics*, vol. 8, no. 10, p. 1756, Oct. 2020, doi: 10.3390/math8101756.

[13]. S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," *Annals of Data Science*, vol. 11, no. 1, pp. 217–242, Feb. 2024, doi: 10.1007/s40745-022-00379-8.

[14]. I. T. Julianto, D. Kurniadi, M. R. Nashrulloh, A. Mulyani, and J. I. Komputer, "COMPARISON OF CLASSIFICATION ALGORITHM AND FEATURE SELECTION IN BITCOIN SENTIMENT ANALYSIS," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 3, 2022, doi: 10.20884/1.jutif.2022.3.3.343.

[15]. X. Xiong, X. Guo, P. Zeng, R. Zou, and X. Wang, "A Short-Term Wind Power Forecast Method via XGBoost Hyper-Parameters Optimization," *Front Energy Res*, vol. 10, May 2022, doi: 10.3389/fenrg.2022.905155.

[16]. K. Riehl, M. Neunteufel, and M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation," Jun. 2023, doi: 10.1093/jrsssc/qlad057.

[17]. Kurnia D, Mazdadi M, Kartini D, Nugroho R, and Abadi F, "SELEKSI FITUR DENGAN PARTICLE SWARM OPTIMIZATION PADA KLASIFIKASI PENYAKIT PARKINSON MENGGUNAKAN XGBOOST," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 5, pp. 1083–1094, Oct. 2023.