Detecting Phising Website using Machine Learning Methods

Ghulam Alvi Rahmat
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
6025232023@student.its.ac.id

Riyanarto Sarno
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
riyanarto@its.ac.id

Kelly Rossa Sungkono
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
kelly@its.ac.id

Agus Tri Haryono
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
7025231020@student.its.ac.id

Abdullah Faqih Septiyanto
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
7025221022@student.its.ac.id

Sholiq
Faculty of Intelligent Electrical and Informatics Technology
Institut Sepuluh Nopember
Surabaya, Jawa Timur, Indonesia
sholiq@its.ac.id

2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI) | 979-8-3315-0861-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICoCSETI63724.2025.11019127

***Abstract*— Modern phishing is a sophisticated cyberattack. It steals login details and payment information by posing as a trusted entity. With our increasing dependence on digital platforms, the frequency and sophistication of phishing attacks have escalated. Many studies have examined phishing detection. Most prioritize metrics like accuracy, precision, and recall. However, a significant research gap exists as previous studies have primarily focused on accuracy and evaluation metrics without incorporating validation processes to verify whether the developed models are adequately robust and reliable. Harnessing the power of machine learning, we can classify websites into authentic or fraudulent categories, proposing a robust defense against these malicious schemes. The purpose of this study is to present the novelty of a concise summary of techniques for detecting phishing and to create a framework that can be used to detect phishing. The dataset comprises 662,590 entries and 9 features. This study implements three supervised learning models: Decision Tree, K-Nearest Neighbors (KNN), and XGBoost algorithms. These algorithms were chosen for their dataset knowledge and applicability. According to experiments, the Decision Tree model has the lowest accuracy at 88.66% and the KNN model the highest at 88.94%. The XGBoost model records an accuracy of 90.24%. XGBoost often achieves high accuracy due to its gradient boosting framework, which combines multiple decision trees to minimize errors. Its regularization techniques also help prevent overfitting, leading to robust performance on unseen data.**

Keywords*— *websites, machine learning, supervised learning, phishing

I. INTRODUCTION

Hackers have used phishing for a long time since 1990s and is still one of the most common and harmful types of cybercrime. Methods used in phishing attacks have changed a lot over the years and now include many different types, such as email phishing scams, spear-phishing, and hacking. In phishing attacks, bad people pretend to be trustworthy entities, like real people or organizations, to get people to give up private information like financial details, including credit/debit card numbers, login credentials, and personal identification [1]. The stolen information is then used to get into private accounts without permission, which can cause financial loses or identity theft. Phishing attacks can also put software on the devices of their victims, which makes security even worse.

This day and age, the internet is an important part of our daily shopping, paying bills, topping up cell phones, and banking lives because so much of our work and personal life is done online. Digital networks make it easier for deals and activities to be carried out quickly in many areas. The internet is essential for because it is so convenient. Users can connect to neighborhood networks and the internet from almost anywhere at any time thanks to improvements in mobile and wireless technologies. Even though digital platforms are useful, our growing dependence on them has also revealed major security holes [2].

Extensive research explores machine learning applications for phishing emails, mostly with the goal of making the detection more accurate and finding the best evaluation measures. Even though these efforts have helped a lot in finding phishing attacks, there is still a big gap in the research. To be more specific, not enough effort is being put into making phishing detection tools that are useful and can be easily added to normal activities. Strong phishing detection systems that work in real time and can be used in real life are needed right now to protect users from phishing risks right away. People often have more than one account on different websites, like bank institutions, social networking sites, and email services. Because they don't know how important their personal information is, online users who aren't paying attention are easy targets for phishing attacks. Phishing scams usually trick people into clicking malicious links by manipulating them with deceptive messages. These links lead to counterfeit websites. These counterfeit websites are usually crafted to closely resemble the legitimate ones. Phishing links are often distributed via popular websites or directly through email to the victim. This redirection misleads the victim into interacting with the attacker's server instead of the authentic one.

Machine learning's application in phishing website detection provides an adaptive solution, unlike traditional rule-based systems [3]. It identifies complex patterns in website features to effectively detect sophisticated phishing attempts. Machine learning excels at recognizing patterns and anomalies in large datasets. By training on a diverse range of legitimate and phishing websites, these models can discern the subtle features that differentiate phishing sites from. This study will take a close look at all the current methods and studies that have been done on phishing attacks. Our focus is

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.

on identifying phishing website URLs by employing three supervised learning algorithms: K-Nearest Neighbors (KNN), XGBoost, and Decision Tree (DT). Then, the model was developed to detect phishing by simply inputting the URL, using the pre-built model for detection. The research provides valuable insights for both researchers and practitioners. For researchers, it delivers a comparative analysis of supervised learning techniques in web security, enhancing their understanding of model performance and feature significance. This contributes to the broader knowledge base in cybersecurity by highlighting the effectiveness of various machine learning methods. For practitioners, the models developed in this study offer practical solutions to strengthen security measures against phishing attacks, thereby safeguarding users' sensitive information. By implementing these models, practitioners can enhance their existing security protocols and reduce the risk of phishing incidents. Furthermore, with improved detection capabilities, organizations can accelerate their incident response times, mitigating potential damage from phishing attacks.

II. Related Work

This section discusses about study that has been done on finding phishing websites as shown in Table I, building on recent reviews and cutting-edge work [4]. Justin et al. [5] showed that using classification models and machine learning, they can find malicious sites better than blacklisting approaches. The normal website data they used came from DMOZ and Yahoo, while the phising website they used came from Spam scatter and Phishtank. They tested Naive Bayes, Support Vector Machine (SVM), and Logistic Regression. Of these, logistic regression worked the best. It had the best automatic feature selection (95% to 99% accuracy) and made a linear representation of the training dataset that was easy to understand.

Godfrey et al. [6] used supervised machine learning, specifically the XGBoost algorithm, to classify phishing URLs. They extracted features from URL properties using datasets from Phish Tank and the University of New Brunswick. Their XGBoost model achieved 86.6% accuracy in detecting malicious websites, slightly outperforming a Multilayer Perceptron model (86.5%). This demonstrates that machine learning surpasses traditional blacklisting for phishing detection.

Mahajan et al. [7] investigated methods to prevent phishing and sought to identify the most effective machine learning model for detecting phishing websites. They compared different algorithms by looking at how accurate they were and how often they made mistakes (false positives and negatives). This showed how machine learning can fix the problems with traditional blacklist and heuristic methods. They took a dataset of 36,711 URLs from Phishtank and Alexa then experiment with different training-testing data splits: 50/50, 70/30, and 90/10 to optimize the performance. The dataset had 19,653 phishing URLs and 17,058 real URLs. They examined at three machine learning models: Decision Tree, SVM, Random Forest. Random Forest worked the best, with a 97.14% success rate and the lowest rate of false negatives. Their study also showed that bigger training datasets made detection more accurate.

Mohammad et al. [8] made a software tool for feature extraction to help people find phishing emails more accurately and avoid misclassifying malicious websites. Two sets of URLs—2,500 phishing URLs from Phishtank and Millersmiles—and 450 real URLs were used. The decision-tree-based C4.5, the separate-and-conquer RIPPER algorithm, the coverage algorithm PRISM, and the association-rule-based CBA were all put to the test. The best results came from CBA, which had a 4.75% mistake rate.

Rathod et al. [9] used machine learning algorithms: LightGBM, XGBoost, Random Forest, CNN to classify phishing URLs. Trained on 651,191 URLs, these models accurately detected malicious URLs, including phishing sites. Hyperparameter tuning improved performance. The ensemble model, combining individual model predictions, achieved the highest accuracy (92.5%) and a precision of 89%.

Lakshmi et al. [10] investigated the application of machine learning to identify phishing websites. Using external resources like blacklists and search engines to make guesses more accurate was a key part of their method for finding things. They used supervised learning algorithms like Naïve Bayes, Decision Tree, Multilayer Perceptron to train classification models on a set of 100 real websites and 100 phishing websites. Tenfold cross-validation was used to test how stable the models were. It turned out that the decision tree approach worked the best.

Akanbi et al. [11] developed a novel method for identifying phishing websites through machine learning techniques. To help the algorithm correctly understand and sort the URLs, they pre-processed the data using methods such as extract the features, dataset splitting, normalization, and importance assigned to attributes. There were phishing and real website names in the dataset. Training and testing datasets were experimented using 50/50, 70/30, and 30/70 splits. They tried several machine learning algorithms, such as K-Nearest Neighbors, Decision Tree C4.5, Linear Regression, Support Vector Machine, to see how well they did at classifying data. After training, pessimistic prunning was used on the Decision Tree to prevent overfitting. The data showed that pruning decision trees makes it easier to spot phishing websites.

Shahrivari et al. [12] did a study that compared different machine learning methods for finding phishing websites. They used a PhishTank collection of about 11,000 sites, with about 4,900 phishing and 6,200 real ones, and found that Random Forest and XGBoost were the fastest and most accurate. Alam et al. [13] used a Kaggle dataset to test Random Forest and Decision Tree models for finding phishing attacks. Using a confusion matrix, they chose 32 key traits and found that Random Forest was 97% accurate. Data on phishing websites from the UCI Machine Learning Repository was used in another work by Yahya et al. [14] to test KNN, Decision Tree, and Random Forest. They started with 32 features and used the Boruta feature selection method to get it down to 15. Then they found that KNN gave the best results.

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.

Table I. Comparative Analysis of Machine Learning Techniques for Phishing Website Identification

Author	Machine learning model	Outline
[5]	Logistic Regression, SVM, Naïve Bayes (NB)	– Look at suspicious URLs to find dangerous websites. – The DMOZ Open Directory Project and Yahoo Directory are the source of the normal dataset. PhishTank and Spam Scatter are sources of the phishing dataset. – Logistic regression automates feature selection very well and gets classification accuracy of 95% to 99%.
[6]	Multilayer Perceptron, XGBoost	– Using Phish Tank and UNB datasets. – XGBoost model achieved 86.6% accuracy in detecting phishing websites, exceeding a Multilayer Perceptron model (86.5%) and showcasing machine learning's superiority over traditional blacklisting.
[7]	Random Forest, Decision Tree, SVM	– Machine learning is used to classify phishing websites. – A set of 36,711 URLs is used, with 50:50, 70:30, and 90:10 rates (17,058 safe URLs and 19,653 phishing URLs). – Random Forest is the best model, it was 97.14% accurate and had very few false negatives.
[8]	CBA, C4.5, PRISM, RIPPER	– Use software to correctly spot phishing sites. This tool has a lot of features that make it less likely that it will miss real phishing sites. – 2,500 phishing website addresses from PhishTank and Millersmiles and 450 real website addresses were used to train the software. – The CBA method, which was the best, had the lowest error rate, at 4.75%.
[9]	LightGBM, XGBoost, Random Forest, CNN	– Machine learning models (LightGBM, XGBoost, Random Forest, CNN) trained on 651,191 URLs achieved 92.5% accuracy and 89% precision in detecting phishing URLs, with ensemble methods showing best performance.
[10]	Naïve Bayes (NB), Decision Tree Induction, Multilayer perceptron (MLP)	– Use machine learning to find phishing websites quickly and easily. – Tenfold cross-validation was used to look at 100 real websites and 100 phishing websites. – The Multilayer Perceptron (MLP) model achieved an accuracy of 97%, outperforming the Naïve Bayes (NB) model, which recorded an accuracy of 93.5%.
[11]	KNN, Linear Regression, Decision Tree C4.5, SVM	– The Decision Tree C4.5 demonstrated exceptional performance with an accuracy of 99.71%. – The datasets splitted in three ways: 50/50, 70/30, and 30/70. – A pruned decision tree can simplify the phishing website classification process.
[12]	KNN, Decision Tree, Logistic Regression, SVM, AdaBoost, Random Forest, Gradient Boosting, XGBoost, Artificial Neural Networks	– Used several machine learning methods that can identify phishing websites. – Use a dataset of 11,000 sites (4,898 phishing and 6,157 real) from PhishTank. – Both Random Forest and XGBoost algorithms efficiently and accurately processed the data, with Random Forest achieving an accuracy of 97.27% and XGBoost outperforming it with an accuracy of 98.32%.
[13]	Random Forest, Decision Tree	– Use the Decision Tree and Random Forest methods to find phising attacks. – The Random Forest achieved 97% accuracy.
[14]	Decision Tree, K-Nearest Neighbor, Random Forest	– Classify phishing attacks using the K-Nearest Neighbor, Decision Tree, Random Forest. – Using Phishing Website Data from the UCI Machine Learning Repository. – Random Forest proved most accurate, correctly identifying 97.69% of the test samples.

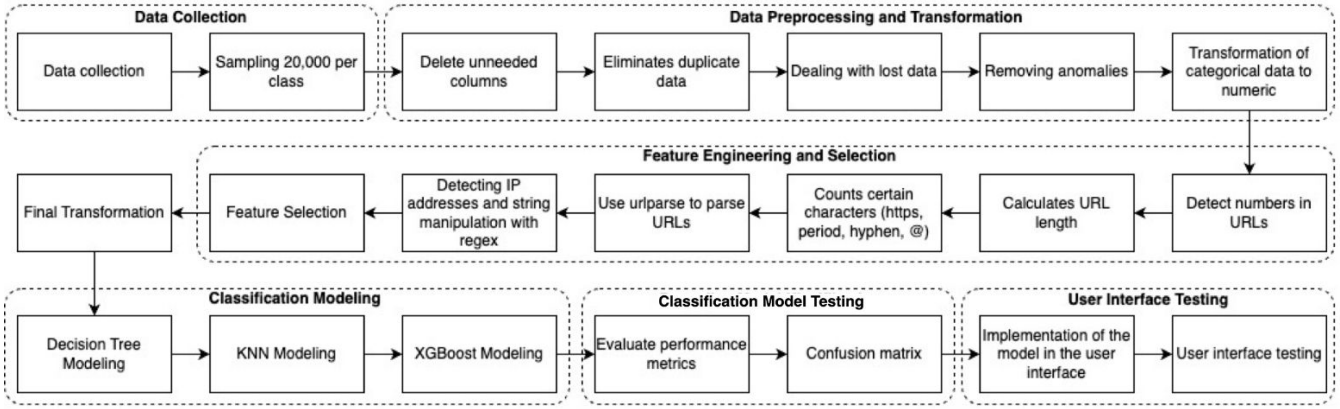

Fig. 1. Diagram of a complete research methodology phishing classification system using machine learning.

III. Proposed Method

The diagram of the research methodology is shown in Fig. 1. The process begins with data collection, where 20,000 samples per class are gathered, followed by data preprocessing steps including removing unnecessary columns, eliminating duplicates, handling missing data, removing anomalies, and converting categorical data to numeric formats [15]. The feature engineering phase involves URL analysis, including detecting numbers in URLs, calculating URL lengths, counting specific characters, parsing URLs with urlparse, and detecting IP addresses using regex. After feature selection and final transformation, the classification modeling phase

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.

employs three models. We compared the eager and lazy ways of learning using Decision Tree, KNN, and XGBoost models. Finally, the system undergoes performance evaluation through performance metrics and confusion matrix, concluding with implementation and testing of the user interface.

This research is based on a publicly available dataset of website URLs, which forms the foundation of the entire process [16]. A big phishing dataset was curated by S. Anjali from Kaggle [17] was used in this study. It has 331,295 records and 10 features. This dataset labels phishing and real websites by using distinctions identified in prior analyses of legitimate and phishing websites, rather than questionnaires. To ensure balanced representation, the imported dataset was sampled to 20,000 instances per class [18].

The training part of the prepared dataset was used to train Decision Tree, K-Nearest Neighbors (KNN), and XGBoost how to classify the URL. A confusion matrix was used to carefully look at performance. We saved the best machine learning model learned on our dataset using the pickle module.

We can use this matrix to figure out how accurate the Decision Tree, KNN, and XGBoost models were, which shows how well each one did [19]. We then assess the accuracy by dividing the number of correct predictions by the total prediction using Eq. (1). This gives a full picture of how well each model can tell the difference between phishing sites and real ones.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

From the confusion matrix, we can figure out error rate, sensitivity (also called recall), specificity, and precision, among other common metrics. The error rate in Eq. (2) is the proportion of inaccurate predictions to data points. Reduced errors indicate a more accurate data classification and prediction model.

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \quad (2)$$

To find the sensitivity, we divide by splitting true positive cases identified correct prediction by the total number of real positives data in the dataset using Eq. (3). One important way assess a classification model's performance is to look at its sensitivity, which is also known as its recall or true positive rate. This metric checks how well the model finds the positives can correctly spot positive instances, which in the case of phishing detection means how well it finds phishing websites. It is very important to have a good sensitivity number so that the model can discover phishing websites and not miss any attempts to steal the information of the user. This lowers the chance that any phishing attempts will go unnoticed.

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

Specificity is calculated by Eq. (4) which divide the number of properly identified correct negative cases by the total number of negative cases in the dataset. This number, which is also called the "true negative rate," is very important to measure how well a classification model works.

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

Eq. (5) determines precision as the ratio of correctly detected positive cases to expected positive cases [20]. Classification model evaluation requires this measure, also known as positive predictive value. Precision is crucial in cybersecurity, where false positives are costly. Phishing detection model can give more reliable and accurate results by keeping a high level of precision. This makes total cybersecurity measures better. Precision is crucial in cybersecurity, where false positives are costly.

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

IV. Result and Discussion

A. Decision Tree: The First Experiment

Experiment one employs a Decision Tree algorithm. This is a supervised method for classifying or regressing outcomes. The goal is to learn easy decision rules that take into account all the variables and put URLs into some groups. This study used scikit-learn's DecisionTreeClassifier package. This data cleaning method identified missing, duplicate, unusually large, or unusually narrow values in the dataset. The data was be shuffled before partitioning. Eighty percent of the data becomes the training set, with the remaining 20% used for testing. The efficacy of the first experiment is assessed using Eq. (1), which calculates the Decision Tree model's accuracy. Table II shows that the model was accurate about 88% of the time, which means that the goal of the experiment, which was to properly classify website types, was met. The confusion matrix is displayed in Fig. 2, which shows how the true positives, true negatives, false positives, false negatives are spread out [21]. This gives us more information about how well the model works.

B. K-Nearest Neighbor: The Second Experiment

The KNN (K-Nearest Neighbors) method is flexible and can be used to solve many supervised learning problems, such as security cases. The KNN method has been used in several studies to categorize fake websites used for phising. In KNN, similar elements are grouped together [22]. This non-parametric classification algorithm generalizes when it needs to classify an instance. Because of its adaptability, the KNN method can handle a wide range of supervised learning challenges, including security-related ones [23]. Table II shows that, using the experimental model, a KNN model with K=15 achieved the highest accuracy, approximately 88.94%.

Table II. Comparison of Evaluation Metrics for Each Algorithm

	Decision Tree	*KNN*	*XGBoost*
Accuracy	88.66%	88.94%	90.24%
Error Rate	11.34%	11.06%	9.76%
Sensitivity	88.66%	88.94%	90.24%
Specificity	98.60%	98.67%	98.99%
Precision	88.86%	89.20%	90.55%

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.

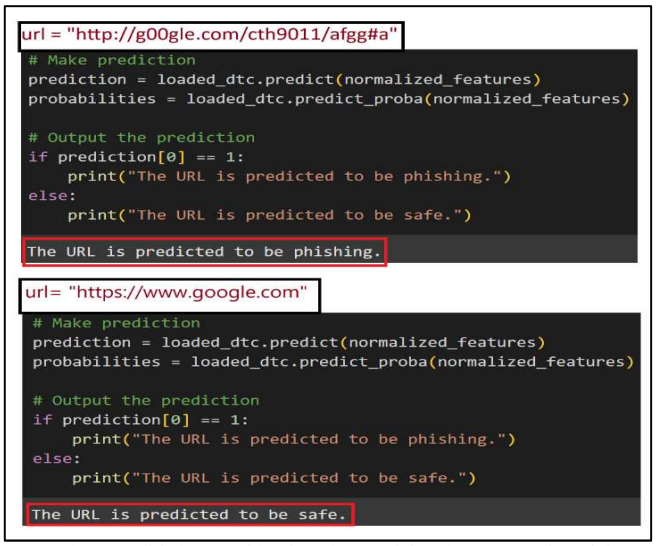

Fig. 3. Validating machine learning models using exported models.

C. *XGBoost: The Third Experiment*

The XGBoost algorithm is used to run the third experiment. XGBoost is a more powerful and customized version of Gradient Boosting that is made to improve speed and performance. Its success depends on how well it can be used in different situations. Out-of-core computation in XGBoost let data scientists handle datasets with hundreds of millions of instances on the desktop. Ultimately, the combination of these techniques creates an end-to-end system that can scale to even larger datasets with minimal cluster resources [24]. Table II shows that XGBoost is accurate about 90.24% of the time based on the plan for the experiment that was made. When comparing our proposed methodology's performance with existing approaches, it is important to consider the methodological differences between studies. Our models achieved accuracy rates from 88.66% to 90.24%. These values are lower than the over 95% accuracy in some prior research due to differences in experimental design. Considering these methodological nuances, our XGBoost model, with 90.24% accuracy, demonstrates robust performance, characterized by a high specificity of 98.99%, showing its excellent ability to correctly identify legitimate instances. This is coupled with a balanced precision of 90.55%, indicating reliable positive predictions. These combined metrics highlight our model's practical applicability, minimizing false positives, which is crucial for real-world deployment.

D. *Phising Detection Interface*

The models that have been built will then be saved using 'pickle'. Python's pickle library is used for serializing and deserializing objects. Python objects undergo transformation into a byte sequence by serialization, or pickling, facilitating storage in files or transfer across networks. The opposite process, reconstructing a Python object from a byte stream, is called deserialization (or unpickling) [25].

By putting a URL within a machine learning algorithm through an easy-to-use Python interface, it is possible to tell if the URL is a phishing attempt as seen in Fig. 3. Users can put in URLs and get instant feedback on how trustworthy thee URLs are. String extraction methods are used to break down the URL so that it can be analyzed. Things that are looked at are the total length, the number of dots (which show possible subdomains), and the path depth (which shows how complicated the URL is). There is also a look at the number of dashes (which are often used in phishing URLs), the appearance of "@" (which could send the browser somewhere else), and the use of an IP address rather than name of the website. Other things are also taken into account, such as the length of the path, the use of numbers, and HTTPS. To find phishing URLs, the Python interface changes the URL based on these extracted traits and sends it to the machine learning model. The analysis focuses on publicly accessible URL characteristics. However, it's important to acknowledge that the effectiveness of the developed models might vary across different geographical regions or internet landscapes. Phishing tactics and the prevalence of certain URL patterns can differ. While the features used in this study are generally applicable, future research could investigate region-specific URL features and evaluate model performance across diverse internet environments to assess the need for regional adaptations or model retraining.

There were two tests in this experiment: one category identified as potentially phishing and the other verified as real. The main objective was to test how well the machine learning model could correctly spot harmful URLs by looking at different parts of them. The objective was to assess the model's capability to detect phishing attempts and to ensure the robustness of the system in flagging potentially harmful URLs. More tests were done on the system using real URLs to see how well it could find real websites and not mistakenly mark them as phishing attempts. This important step made sure that the machine learning model could reliably tell the difference between URLs that are dangerous and those that are not.

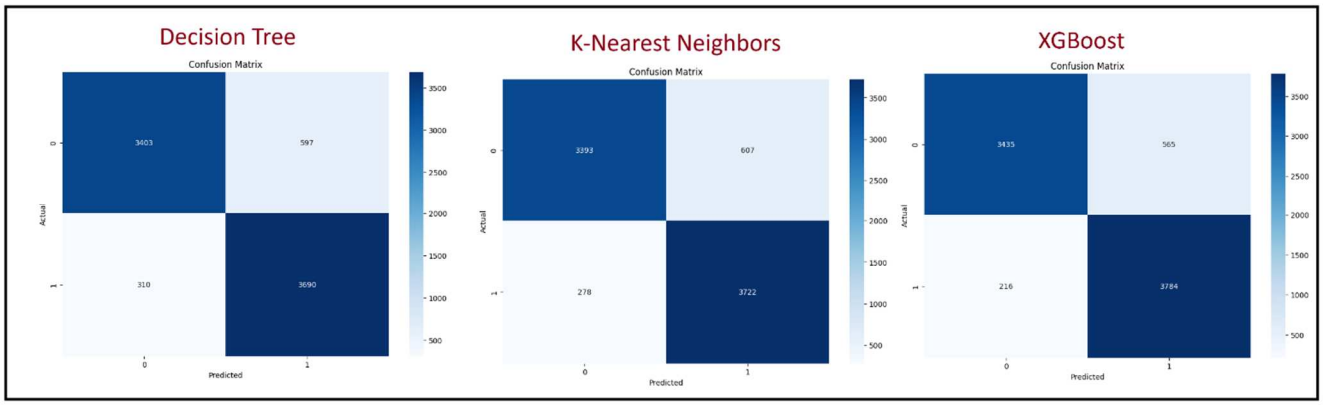

Fig. 2. Comparison of confusion matrices for each model.

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion

We successfully achieved its objective of developing a model to detect URLs considered as phishing. Pickle was suggested as a way to store the learned model and then load it again, making it useful in a variety of situations. This method makes sure that the model can be used effectively without having to be trained again. This lets it be added to real-time hacking detection systems on a number of different platforms. As part of the study, several machine learning models were developed and tested to see how well they could spot phishing URLs. The evaluation results we conducted show that among the various algorithms tested, XGBoost had the highest accuracy (90.24%), which is a great result. With 88.66% accuracy, the Decision Tree model did the worst. KNN did a little better, with an 88.94% score. As these results show, XGBoost is good at finding phishing URLs, which makes it a useful security tool.

Future research should explore the incorporation of additional features, such as website content analysis to enhance the model's accuracy. Exposing it to larger and more varied datasets could help it detect more types of hacking attempts. Integrating the model into real-time phishing detection tools would provide immediate protection against phishing.

Author Contributions

Ghulam Alvi Rahmat performed the research, experiments, data analysis, and manuscript drafting. Riyanarto Sarno conceived the idea, supervised, guided, and revised the manuscript. Agus Tri Haryono preprocessed data and trained the model. Kelly Rossa Sungkono provided machine learning expertise and helped design the research. Abdullah Faqih Septiyanto assisted with literature review and data analysis. Sholiq coordinated the project, facilitated communication between team members, and ensured timely progress.

Acknowledgment

This research was funded by Institut Teknologi Sepuluh Nopember (ITS) under Penelitian Kolaborasi Pusat and under the Project Scheme of the Publication Writing and Intellectual Property Rights (IPR) Incentive (Penulisan Publikasi dan Hak Kekayaan Intelektual/PPHKI) Program 2024

References

[1] Z. Alkhalil, C. Hewage, L. Nawaf and . I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," Frontiers in Computer Science, 2021.

[2] E. Overby, S. A. Slaughter and B. Konsynski, "The Design, Use, and Consequences of," Information Systems Research, vol. 21, pp. 700-710, 2010.

[3] V. Borate, A. Adsul, R. Dhakane, S. Gawade, S. Ghodake and P. Jadhav, "A Comprehensive Review of Phishing Attack Detection Using Machine Learning Techniques," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 4, no. 3, 2024.

[4] L. Tang and . Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing," Machine Learning & Knowledge Extraction, vol. 3, pp. 672-694, 2021.

[5] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, "Identifying Suspicious URLs: An Application of Large-Scale Online Learning," Proceedings of the 26th annual international conference on machine, pp. 681-688, 2009.

[6] O. C. Godfrey, Y. G. Fwa and A. N. Edmund, "Phishing URL Detection: A Basic Machine Learning Approach," International journal of science & technoledge, 2024.

[7] R. Mahajan and I. Siddavatam, "Phishing Website Detection using Machine Learning," International Journal of Computer Applications, vol. 181, no. 3, pp. 45-47, 2018.

[8] R. M. Mohammad, F. Thabtah and L. McCluskey, "Intelligent rule-based phishing websites classification," IET Information Security, 2013.

[9] S. Rathod, N. Panchal, A. Sarowa and V. Chavhan, "SaleSurf - Strengthening the Web's Armour, One URL at a Time - Malicious URL Detection using Machine Learning Models," International Journal For Multidisciplinary Research, 2024.

[10] S. Lakshmi and V. MS, "Efficient prediction of phishing websites using supervised," International Conference on Communication Technology and System Design, 2011.

[11] O. Akanbi, A. Zainal and A. Abunadi, "Phishing Website Classification: A Machine Learning Approach," Journal of Information Assurance and Security, vol. 9, pp. 222-234, 2014.

[12] V. Shahrivari, M. M. Darabi and M. Izadi, "Phishing Detection Using Machine Learning," 2020.

[13] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R. E. Ulfath and S. Hossain, "Phishing Attacks Detection using Machine Learning Approach," 2020 Third International Conference on Smart Systems and Inventive Technology, 2020.

[14] F. Yahya, . R. I. W. Mahibol , C. K. Ying, M. B. Anai, S. A. Frankie, E. L. N. Wei and R. G. Utomo, " Detection of Phising Websites using Machine," 2021 International Conference on Data Science and Its Applications, 2021.

[15] A. Sutranggono, R. Sarno and I. Ghozali, "Multi-Class Multi-Level Classification of Mental Health Disorders Based on Textual Data from Social Media," Journal of Information and Communication Technology, vol. 23, no. 1, pp. 77-104, 2024.

[16] M. Yamin and R. Sarno, "Hybrid neural machine translation with statistical and rule-based approach for syntactics and semantics between Tolaki-Indonesian-English languages," Periodicals of Engineering and Natural Sciences, vol. 11, no. 5, pp. 117-136, 10 2023.

[17] S. Anjali, "Kaggle," 2024. [Online]. Available: https://www.kaggle.com/datasets/simaanjali/phising-detection-dataset.

[18] Salsabila, R. Sarno, I. Ghozali and K. R. Sungkono, "Improving cyberbullying detection through multi-level machine learning," International Journal of Electrical and Computer Engineering (IJECE), vol. 14, no. 2, pp. 1779-1787, 2024.

[19] M. Malikhah, R. Sarno and S. I. Sabilla, "Ensemble Learning for Optimizing Classification of Pork Adulteration in Beef Based on Electronic Nose Dataset," International Journal of Intelligent Engineering and Systems, vol. 14, no. 4, pp. 44-55, 2021.

[20] I. Ghozali, K. R. Sungkono, R. Sarno and R. Abdullah, "Synonym based feature expansion for Indonesian hate speech detection," International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 1, 2023.

[21] J. Erbani, P.-É. Portier, E. Egyed-Zsigmond and D. Nurbakova, "Confusion Matrices: A Unified Theory," IEEE Access, vol. 12, pp. 181372-181419, 2024.

[22] M. Jupri and R. Sarno, "Taxpayer Compliance Classification Using C4.5, SVM, KNN, Naive Bayes and MLP," in International Conference on Information and Communications Technology (ICOIACT), 2018.

[23] F. Toolan and J. Carthy, "Feature selection for Spam and Phishing detection," 2010 eCrime Researchers Summit, 2011.

[24] . T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794, 2016.

[25] "Docs Python," [Online]. Available: https://docs.python.org/3/library/pickle.html.

Authorized licensed use limited to: Institut Teknologi Sepuluh Nopember. Downloaded on October 02,2025 at 14:19:10 UTC from IEEE Xplore. Restrictions apply.