International Journal on Advanced Science Engineering Information Technology

Vol.15 (2025) No. 1
ISSN: 2088-5334

Phishing Domain Detection Using Machine Learning Algorithms

Dinny Komalasari [a,*], Tri Basuki Kurniawan [b], Deshinta Arrova Dewi [c], Mohd Zaki Zakaria [d], Zubaile Abdullah [e], Alde Alanda [f]

[a] *Faculty of Vocasional, Universitas Bina Darma, Palembang, Indonesia*
[b] *Postgraduate Program, Universitas Bina Darma, Palembang, Indonesia*
[c] *Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia*
[d] *Faculty of Computer & Mathematic Sciences, University Technology Mara, Malaysia*
[e] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia*
[f] *Department of Information Technology, Politeknik Negeri Padang, Padang, Indonesia*
Corresponding author: [*] dinny.komalasari@binadarma.ac.id

***Abstract*— Phishing, a prevalent cyber threat, continues to jeopardize sensitive information by exploiting the vulnerabilities of digital platforms. This research investigates the escalating danger of phishing attacks, focusing on the creation of deceptive websites known as phishing domains. Leveraging machine learning algorithms, particularly supervised and unsupervised learning techniques, the study aims to proactively identify and classify these malicious domains by analyzing diverse factors like domain names, online content, SSL certificates, and historical data. The proposed solution involves the development of prediction models using decision trees, random forests, support vector machines, and Gradient Boosting, with the latter exhibiting the highest accuracy at 92%. The system assigns risk scores to domains based on properties such as registration details and SSL certificate validity, facilitating the real-time identification of potential phishing activities. The research addresses the critical need for data security in the face of phishing threats affecting individuals and businesses, providing a robust defense mechanism against evolving cyber threats. Recommendations for continuous model training, regular updates, diversification of dataset sources, and integration with existing security infrastructure aim to enhance the system's adaptability and resilience in countering emerging phishing threats. Overall, this study contributes to ongoing efforts in cybersecurity, offering a proactive defense mechanism against the pervasive and evolving challenges posed by phishing attacks.**

***Keywords*— Phishing detection; machine learning; cybersecurity; domain analysis; threat prevention; process innovation.**

Manuscript received 4 Sep. 2024; revised 17 Nov. 2024; accepted 10 Jan. 2025. Date of publication 28 Feb. 2025.

I. INTRODUCTION

Phishing is the most basic type of cybercrime that aims to lure individuals into revealing sensitive information, including personally identifiable data, banking and credit card particulars, login credentials, and passwords [1], [2]. The credentials or private information stolen are then utilized to access the victims' essential records, which can lead to significant fraud and financial loss [3], [4]. Phishing attacks have developed into a pervasive hazard because of the rising reliance on digital platforms and the extensive acceptance of online services, preying on the vulnerabilities of unwary consumers. Attackers are now using more advanced strategies as classic phishing techniques develop, such as creating duplicitous websites known as “phishing domains” [5].

Researchers and cybersecurity experts have used machine learning algorithms to tackle this problem since they have shown promise in locating and classifying phishing domains according to their traits and patterns. Machine learning algorithms can be taught to distinguish between trustworthy and malicious domains by examining various factors, including domain names, online content, SSL certificates, and historical data.

The most efficient and standard method for detecting phishing domains is using Machine Learning algorithms for classification [1], [6], [7]. Due to its capacity to analyze vast amounts of data and spot patterns that might separate trustworthy websites from phishing ones, machine learning algorithms have attracted much interest in cybersecurity. These algorithms may extract valuable information and provide reliable models for precise phishing domain identification by utilizing a variety of variables, including

domain properties, content analysis, and user behavior. The success of this strategy is highlighted by Pascariu and Bacivarov [1], who emphasize that it effectively counters phishing assaults' tendency to evolve.

In conclusion, machine learning algorithms for phishing domain recognition offer a viable strategy for thwarting phishing attempts [8]. These algorithms provide a proactive defense mechanism against developing cyber threats because they can analyze enormous volumes of data and learn from trends [9]. The continuous efforts to protect people, companies, and online ecosystems from the adverse effects of phishing assaults will be aided by further study and development in this area.

Data security has become a significant problem due to the world's fast digitization, and one cyberattack known as phishing is used to steal and use users' personal information. Phishing is one of the most significant risks to information security among the many web application threats in the Internet domain [10]. Recent statistics show that over 33 million records are expected to be extorted by 2023, with a ransomware or phishing attack occurring every 11 seconds. To protect individuals who frequently use email and social media and want to safeguard their personal information from phishing attacks, a potential solution could be to develop prediction models of genuine or malicious domains.

In addition to the risk of personal information being stolen, businesses are also vulnerable to phishing attacks, which can lead to unauthorized access to sensitive information such as financial data and trade secrets [1]. This is a serious security concern that requires a solution. Developing a system that checks whether a website is legitimate or malicious could help businesses avoid phishing scams and protect their sensitive information.

To accomplish this purpose, the proposed system would categorize domains according to their properties using various supervised learning methods, including decision trees, random forests, and support vector machines. These characteristics may include the age of the domain, details about its registration, the validity of its SSL certificate, and linguistic elements taken directly from the domain name. The system can discover possible phishing activity indicators by examining these traits, and it can also give each domain a risk score that indicates how likely it is to be harmful.

In summary, the proposed study seeks to use machine learning algorithms to tackle the urgent problem of phishing domain identification. This study can help to strengthen data security and protect people and organizations from phishing scams by creating an effective and reliable prediction model.

The importance of this study is in demonstrating the urgent need for reliable phishing domain recognition techniques to stop such assaults. This paper aims to support the development of accurate and trustworthy methods for recognizing and blocking phishing domains by reporting the findings of experiments performed to assess the effectiveness of machine learning algorithms. To give insightful information on the efficacy of various detection strategies, it is crucial to highlight the advantages and disadvantages of the employed methodology and compare it to current procedures. The findings of this study highlight the significance of taking preventative action against phishing attacks by having the ability to drastically decrease financial losses and protect businesses and individuals from severe economic harm.

II. Material and Method

A. Phishing Domain Attack

Phishing domains deceive users and trick them into submitting sensitive information, such as their authentication details, including usernames, passwords, and unique codes associated with multifactor authentication. The stolen credentials enable attackers to access legitimate services, impersonating real users and getting access to sensitive information [11]. According to Wazirali *et al.* [12], this attack is a very well-known type of cybercrime. It is straightforward to bait unaware users to click fake websites for some prize and offer rather than attacking the computer defense system.

This is because malicious websites are designed similarly in terms of look and feel, making them seem genuine. El-Rashidy [13] stated that 1,220,523 phishing assaults were detected in 2016, a 65% increase over 2015. These attacks use constantly evolving strategies. In just two years, the attacks rose by 476% over 2018 and 226% over the third quarter of 2019, according to the Anti-Phishing Working Group (APWG) report.

Spoof and concocted websites are two common phishing domain types [14]. A website created to look like the target website is called a spoof website, but it is not as perfect as a legitimate website. Spoof websites usually have bugs and grammatical errors. Also, the visual interface, such as the design layout, is inconsistent. A concerted website is a fraudulent website appearing as a legitimate site, such as a provider of commercial services. For example, they receive customer payment, and the product was never shipped.

B. Phishing Domain Detection Technique

The traditional approach to detecting phishing domains requires manual analysis and automated techniques. Chatterjee and Namin [3] stated that blocklisting is the most famous conventional phishing detection technique. Aljofey et al. [15] said allowlisting is one of the most effective traditional techniques to avoid phishing attacks. Blocklisting involves creating and maintaining a list, often referred to as a blocklist, of domain names and URLs that have been previously identified as phishing domains or associated with fraudulent activities.

Blocking works by keeping a list of known phishing domains or suspicious URLs; it compares the visited domain to this list. This approach typically requires collecting information about reported phishing attacks, analyzing domain characteristics, and monitoring online sources for new phishing attempts. When a user attempts to access a website or click on a link, the domain or URL will be compared to the domain in the blocklist. The domain is marked as possibly harmful if a match is found. Most blocklist methods are widely employed in industry due to low false positives, but denylists alone cannot generalize well to unseen phishing instances [16]. For example, [17] reported that the effort required is too massive to manage because phishing domains have a short lifetime, and new ones are designed quickly.

Machine learning has proven to be a valuable tool in phishing domain prediction [18]. Sharma et al. [19] stated that

artificial intelligence and machine learning technologies have played a significant role in the effectiveness of anti-phishing algorithms. By utilizing large datasets, machine learning approaches can analyze and detect complex patterns and characteristics associated with phishing attacks [20]. Therefore, it can enhance the precision of detecting domains linked to malicious threats.

C. Using Machine Learning Algorithms

In this part, several machine learning algorithms will be discussed to explore how these algorithms might help the detection and prediction of phishing domains. Various machine learning algorithms, such as support vector machines, random forests, naïve Bayes, and decision trees, have been widely studied and applied in anti-phishing [21]. A study from Shieh et al. [22] shows that each algorithm offers unique capabilities and characteristics that can be classified to identify and classify phishing domains accurately.

Support vector machine is one of machine learning technology's most widely used algorithms. Support vector machine algorithm: Each data item is plotted as a point in n-dimensional space, and the support vector machine algorithm constructs a separating line for the classification of two classes; this separating line is well known as a hyperplane [23]. This hyperplane aims to achieve the best possible separation by maximizing the margin between classes.

A support vector machine is also a crucial classifier in the machine learning concept, according to Sahingoz *et al.*[24], which discovers non-linear decision boundaries by utilizing the kernel technique in the training data. A quick training approach for SVMs is Sequential Minimal Optimization (SMO). SMO is one of the most used algorithms for classification issues because it is straightforward. Additionally, it is utilized to resolve optimization issues during training.

Random forest is supervised machine learning. The random forest produces different decision trees. Each tree is constructed using a different bootstrap test and a tree classification method using the initial data [25]. This process introduces randomness and diversity into the ensemble, which helps to reduce overfitting and improve the model's generalization ability [26].

In a random forest, each decision tree separately predicts something, and the final prediction is made by combining the results of all the individual trees. Voting or averages can accomplish this aggregate (for classification or regression tasks) [27]. Therefore, a random forest, as opposed to a single decision tree, can deliver more precise and reliable outcomes by aggregating the predictions of numerous trees.

The Naïve Bayes classification is a probabilistic machine learning algorithm that is both straightforward and impactful. Naïve Bayes is also preferred in many application areas, such as classifying texts and detecting email spam, due to its simplicity, efficiency, and good performance [24], [28]. It is based on the Bayes theorem, which describes the relationship of conditional probabilities of statistical quantities. It assumes independence between the attribute values [29].

The Decision Tree classification is a widely adopted supervised learning technique for classification and regression purposes. Sahingoz *et al.* [24] asserted that the classifier repeatedly partitions the training dataset into subparts, forming a tree-like structure that helps identify separation lines. These lines then determine the appropriate class for a given target item. At each decision node, the data is split into multiple categories based on a specific attribute value. Each leaf node is assigned to a class in the classification algorithm, often by calculating a probability. The decision tree creates a training model to predict the target value or class in the tree representation. Each internal node of the tree belongs to an attribute, and each leaf node belongs to the class label.

In general, the extreme gradient boosting algorithm uses an ensemble approach known as boosting to add new models (decision trees) to fix errors produced by previous models [30]. This repeated process of boosting continues until no more improvements can be made. What separates major gradient boosting is its emphasis on improving the model's overall performance through many essential aspects. It utilizes regularization terms like L1 and L2 to prevent overfitting, parallel processing for quicker calculation on big datasets and effectively handles missing information. Furthermore, extreme gradient boosting employs tree pruning techniques to generate more straightforward and efficient models by deleting unneeded branches. Extreme gradient boosting adaptability extends to its support for alternative loss functions and evaluation criteria, making it suitable for a wide range of machine learning problems such as regression and classification.

D. Methodology

The project was broken down into 8 phases per this research methodology. There were numerous phases, with the preliminary study phase as the first and the documentation phase as the final. The project flow started with the initial study, knowledge acquisition, data collection, data preprocessing, model development, system development, system integration, testing, and evaluation, and ended with documentation. The research design, through which the study objectives were created, was described in depth in the research framework.

1) Preliminary Study

The preliminary study, the first stage of the research project, was vital in laying the groundwork for the complete investigation. Intensive activities were carried out during this period to comprehend the subject area thoroughly. These activities included conducting a detailed background study and examining relevant literature and previous research to establish a strong background. The issue and area of interest were also identified, enabling a precise specification of the study issue and its reach.

The main outputs of this phase included a clearly stated research background, a precisely formulated problem statement, a research question that directed the study, a defined scope that established the parameters of the research, the identification of significant prior research in the field, and a thorough literature review that synthesized knowledge already in existence. These deliveries collectively provided a solid foundation for the subsequent phases of the research project.

2) Knowledge Acquisition

An extensive literature analysis and an examination of the benefits and drawbacks of the most recent prediction models

were part of the knowledge acquisition phase of the research project. This step included tasks such as looking through pertinent sites like IEEE Explore, Scopus, Google, and Elsevier to obtain essential information.

The deliverables of this phase consisted of a research background informed by the literature review, a well-defined problem statement, a research question that directed the study, a clearly defined scope, the identification of significant research in the field, and an extensive literature review that critically evaluated current prediction models. These deliverables gave a detailed overview of the field's current knowledge and knowledge gaps, laying a strong basis for the project's subsequent phases.

3) Data Collection

During the research project's data-gathering phase, acceptable data sources pertinent to the study's goals were carefully chosen. One of the sources considered was Elsevier.com, which offered valuable datasets for analysis. The datasets that closely matched the study's needs were selected from readily available ones, ensuring they included the data needed to answer the research questions. After finding the suitable datasets, they were downloaded and kept in their raw form.

These unprocessed datasets provided the basis for the subsequent phases of the research endeavor, allowing for additional data cleaning, processing, and analysis to yield valuable insights and draw conclusions. The phase of data collecting was crucial in assuring the availability of high-quality data, which was necessary for producing trustworthy and accurate research results.

4) Data Preprocessing

The research project's data preparation phase included crucial data cleansing and separation tasks. The quality and integrity of the acquired dataset were ensured throughout this step by carefully examining it to find and correct any discrepancies, errors, or missing values. By removing noise, outliers, and unnecessary data, data cleaning procedures increased the dataset's dependability for further analysis. Additionally, the dataset was divided into valuable subsets, such as training and testing sets, to speed up the creation and assessment of models. A cleaned and preprocessed dataset, or refined version of the original data, was the deliverable of this phase and was prepared for additional analysis and modelling in later stages of the research project.

5) Model Development

Support Vector Machines (SVM), Random Forest, Naive Bayes, Decision Tree, and Extreme Gradient Boosting were the five machine learning algorithms that were the emphasis of the research project's model creation phase. In this step, the architecture of each model was sketched out to show its components and structure.

6) System Development

The proposed "Machine Learning-Based System for Phishing Domain Detection" was designed and implemented in great detail throughout the system development phase of the research project. The general structure and organization of the system were carefully defined at this phase, considering elements like the required modules, algorithms, and data flow. To create an efficient and aesthetically pleasing interface, insights, best practices, and design ideas were gathered from various sources, such as Stack Overflow, GitHub, and Dribble.

III. Results and Discussion

It consists of data collection results, data pre-processing results, model training, and testing of the Phishing Domain Detection System Prototype.

A. Data Collection Results

This section highlights the findings of the data-collecting phase, giving insights into the datasets chosen and the general process of acquiring helpful information for the project. The findings presented here serve as a guide for the resulting analyses in the following chapters. The primary focus is on the sources used, the nature of the raw datasets acquired, and the early results produced from the data-gathering process. The dataset collected was from the Elsevier website, and the total data consists of 112 features, 96 of which are extracted from the website address itself. In comparison, the remaining 15 features were extracted using custom Python code.

B. Data Pre-processing Results

Despite having 112 features in the dataset, this project does not use all of them. Instead, a careful feature selection approach was used to identify and prioritize the most critical features that significantly assist prediction models for identifying phishing domains associated with phishing activities. The activities in the pre-processing are checking missing values, which the dataset prepared has no null values, data exploration, and feature selection.

C. Support Vector Machine Result

The outcome of employing a Support Vector Machine in the model training will be covered. The red line indicates the training score, while the green line is the cross-validation score. The figure shows that the training score line started at a lower accuracy of 81% and gradually increased until training examples were 30,000 and 82%. The cross-validation score line started at a higher accuracy of 82.1% and slowly decreased but remained at an accuracy of 82%.

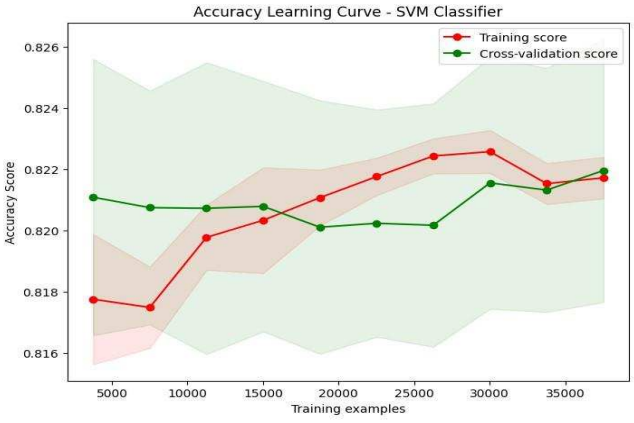

Fig. 1 Accuracy Learning Curve – SVM Classifier (80:20)

For class 0, the model achieved a precision of 0.77, showing that 77% of the instances predicted as class 0 were

correct, and a recall of 0.88, suggesting that 88% of the actual cases of class 0 were successfully identified. The F1-score, a balance of precision and recall, is reported as 0.82 for class 0. Similar metrics are presented for class 1: precision is 0.88, recall is 0.76, and F1-score is 0.81. The model's overall accuracy across both classes is 0.82, implying correct predictions for 82% of the total instances. The macro and weighted averages for precision, recall, and F1 score are also provided, offering a comprehensive summary of the model's performance. These metrics collectively give insights into the model's ability to classify instances and its trade-offs between precision and recall for each class.

TABLE I
CLASSIFICATION SUMMARY – SVM (80:20)

Classification summary – SVM (80:20)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.77	0.88	0.82	5588
1	0.88	0.76	0.81	6141
Accuracy			0.82	11729
Macro Avg	0.82	0.82	0.82	11729
Weighted Avg	0.83	0.82	0.82	11729

For class 0, the model achieved a precision of 0.77, indicating that 77% of the instances predicted as class 0 were correct, and a recall of 0.89, suggesting that 89% of the actual cases of class 0 were successfully identified. The F1 score, a balance of precision and recall, is reported as 0.82 for class 0. Similar metrics are presented for class 1: precision is 0.88, recall is 0.76, and F1-score is 0.81. The model's overall accuracy across both classes is 0.82, implying correct predictions for 82% of the total instances. The macro and weighted averages for precision, recall, and F1-score are also provided, offering a comprehensive summary of the model's performance. These metrics collectively give insights into the model's ability to classify instances and its trade-offs between precision and recall for each class.

In the context of class 0, the precision stands at 0.76, indicating that 76% of instances predicted as class 0 were accurate, while the recall is 0.88, denoting that 88% of actual class 0 cases were correctly identified. The F1-score, representing a harmonized measure of precision and recall, is reported as 0.81 for class 0. Similar metrics are outlined for class 1: precision is 0.87, recall is 0.75, and F1 score is 0.81.

TABLE II
CLASSIFICATION SUMMARY – SVM (70:30)

Classification summary – SVM (70:30)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.77	0.89	0.82	8412
1	0.88	0.76	0.81	9182
Accuracy			0.82	17594
Macro Avg	0.83	0.82	0.82	17594
Weighted Avg	0.83	0.82	0.82	17594

The model's overall accuracy, encompassing both classes, is specified as 0.81, signifying correct predictions for 81% of all instances. Additionally, macro and weighted averages are provided for precision, recall, and F1-score, offering a comprehensive overview of the model's performance. These aggregated metrics provide valuable insights into the model's proficiency in classifying instances, shedding light on each class's trade-offs between precision and recall.

TABLE III
CLASSIFICATION SUMMARY – SVM (90:10)

Classification summary – SVM (90:10)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.76	0.88	0.81	2770
1	0.87	0.75	0.81	3095
Accuracy			0.81	5865
Macro Avg	0.82	0.81	0.81	5865
Weighted Avg	0.82	0.81	0.81	5865

D. Random Forest Result

Initially, the training score and cross-validation accuracy embark on the learning process at a modest 52%. However, their trajectories swiftly ascend, reaching a notable peak of 85%. Following this sharp ascent, the training score and cross-validation accuracy demonstrate stability, maintaining their commendable performance even as the training examples accumulate up to 35,000 instances. This plateauing at a high accuracy level signifies the robustness and proficiency attained by the Random Forest model, showcasing its ability to generalize diverse data well and maintain predictive excellence across a substantial dataset.

For class 0, the model achieved a precision of 0.85, signifying that 85% of the instances predicted as class 0 were accurate, while the recall was 0.84, indicating that the model captured 84% of the actual cases of class 0. The F1-score, a balance of precision and recall, stands at 0.85 for class 0. Similar metrics are reported for class 1: precision is 0.86, recall is 0.87, and F1-score is 0.86. These values collectively suggest a balanced performance between precision and recall for both classes. The support values indicate 5588 class 0 and 6141 instances of class 1 in the dataset. The model's overall accuracy across both classes is 0.85, signifying that the model correctly predicted the class labels for approximately 85% of the total instances. The macro and weighted averages for precision, recall, and F1-score are all 0.85, indicating consistent and well-rounded performance across the binary classification task.

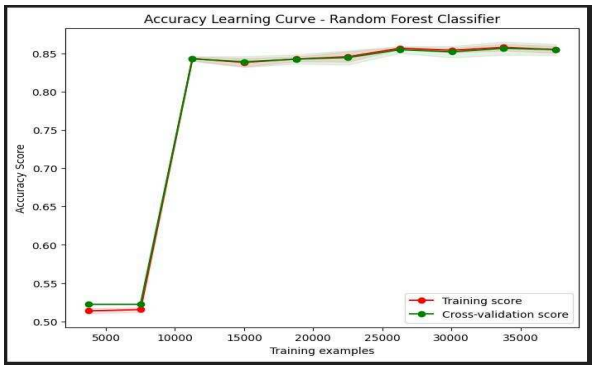

Fig. 2 Accuracy Learning Curve – Random Forest (80:20)

TABLE IV
CLASSIFICATION SUMMARY – RANDOM FOREST (80:20)

Classification summary – Random Forest (80:20)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.85	0.84	0.85	5588
1	0.86	0.87	0.86	6141
Accuracy			0.85	11729
Macro Avg	0.85	0.85	0.85	11729
Weighted Avg	0.85	0.85	0.85	11729

The Random Forest model unfolds a compelling narrative of its training journey. Initially, the training score and cross-validation accuracy embark on the learning process at a modest 52%. However, their trajectories swiftly ascend, reaching a notable peak of 84%. Following this sharp ascent, the training score and cross-validation accuracy demonstrate stability, maintaining their commendable performance even as the training examples accumulate up to 30,000 instances.

Support values further reveal 8,412 class 0 and 9,182 instances of class 1 in the dataset. The model's overall accuracy across both classes is reported as 0.86, indicating that the model correctly predicted the class labels for approximately 86% of the total instances. Notably, macro and weighted averages for precision, recall, and F1-score are reported as 0.86, signifying consistent and well-rounded performance across the binary classification task. This suggests that the Random Forest model maintains a harmonious trade-off between precision and recall, making it a robust choice for accurately classifying instances in the given dataset.

TABLE V
CLASSIFICATION SUMMARY – RANDOM FOREST (70:30)

Classification summary – Random Forest (70:30)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.88	0.82	0.85	8412
1	0.84	0.90	0.87	9182
Accuracy			0.86	17594
Macro Avg	0.86	0.86	0.86	17594
Weighted Avg	0.86	0.86	0.86	17594

Support values further reveal 2770 class 0 and 3095 instances of class 1 in the dataset. The model's overall accuracy across both classes is reported as 0.87, indicating that the model correctly predicted the class labels for approximately 87% of the total instances. Both macro and weighted averages for precision, recall, and F1-score are reported as 0.87, signifying consistent and well-rounded performance across the binary classification task.

TABLE VI
CLASSIFICATION SUMMARY – RANDOM FOREST (90:10)

Classification summary – Random Forest (90:10)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.86	0.85	0.86	2770
1	0.87	0.88	0.87	3095
Accuracy			0.87	5865
Macro Avg	0.87	0.87	0.87	5865
Weighted Avg	0.87	0.87	0.87	5865

E. Naïve Bayes Result

The accuracy learning curve for the Naive Bayes model begins with an initial synchronization between the training score and cross-validation lines, both commencing their journey at an accuracy level of 74.6%. However, an intriguing dynamic follows as the training score experiences a slight descent to 73.5%, subsequently exhibiting a marginal recovery to 74%, only to decline back to 73% gradually. In parallel, the cross-validation score line mirrors a similar pattern, descending from the initial accuracy to 73.4%, rising to 73.5%, and then aligning with the training score by settling at 73%. This oscillation in accuracy levels illustrates the model's adaptive learning process, encountering fluctuations but ultimately maintaining a stable and competitive performance.

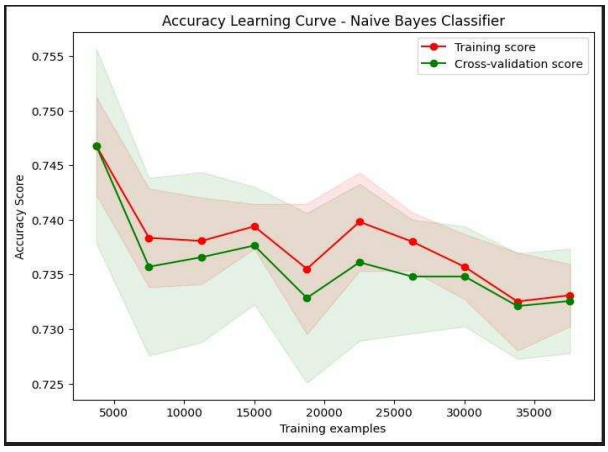

Fig. 3 Accuracy Learning Curve – Naïve Bayes (80:20)

The F1-score, balancing precision and recall, is 0.77 for class 0 and 0.68 for class 1. The model's overall accuracy is 0.73, indicating correct predictions for approximately 73% of the instances. The macro and weighted averages provide additional insights, with macro averaging at 0.78 and weighted averaging at 0.79, offering a comprehensive assessment of the model's performance across both classes.

TABLE VII
CLASSIFICATION SUMMARY – NAÏVE BAYES (80:20)

Classification summary – Naïve Bayes (80:20)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.65	0.94	0.77	5588
1	0.91	0.54	0.68	6141
Accuracy			0.73	11729
Macro Avg	0.78	0.74	0.72	11729
Weighted Avg	0.79	0.73	0.72	11729

This oscillation in accuracy levels illustrates the model's adaptive learning process, encountering fluctuations while maintaining a stable and competitive performance. The observed nuances in the learning curve reveal the model's ability to navigate varying complexities within the training data, showcasing its resilience in achieving consistent accuracy levels despite temporary fluctuations.

TABLE VIII
CLASSIFICATION SUMMARY – NAÏVE BAYES (70:30)

Classification summary – Naïve Bayes (70:30)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.65	0.95	0.77	8412
1	0.92	0.53	0.67	9182
Accuracy			0.73	17594
Macro Avg	0.78	0.74	0.72	17594
Weighted Avg	0.79	0.73	0.72	17594

The classification summary for the Naive Bayes model reveals a nuanced performance across the binary classification task, distinguishing between classes 0 and 1. For class 0, the model exhibits a lower precision of 0.64, indicating that 64% of the instances predicted as class 0 were accurate. However, it demonstrates a high recall of 0.94, capturing 94% of the actual cases of class 0. In contrast, for class 1, the model achieves a higher precision of 0.91,

signifying that 91% of instances predicted as class 1 were correct. Still, it struggles with a lower recall of 0.53, identifying only 53% of the actual cases of class 1.

The F1-score, balancing precision and recall, is 0.76 for class 0 and 0.67 for class 1. The model's overall accuracy is 0.73, indicating correct predictions for approximately 73% of the instances. The macro and weighted averages provide additional insights, with macro averaging at 0.78 and weighted averaging the same as macro averaging, which is 0.78.

TABLE IX
CLASSIFICATION SUMMARY – NAÏVE BAYES (90:10)

Classification summary – Naïve Bayes (90:10)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.64	0.94	0.76	2770
1	0.91	0.53	0.67	3095
Accuracy			0.73	5865
Macro Avg	0.78	0.74	0.72	5865
Weighted Avg	0.78	0.73	0.72	5865

F. Decision Tree Result

The decision Tree model starts with a consistent starting point, as both the training and cross-validation score lines initiate their trajectory at an accuracy level of 53%. This stability persists until reaching 70,000 training examples, where an abrupt and simultaneous ascent occurs. At the 10,000 training examples mark, the accuracy spikes to an impressive 84%, indicating a substantial improvement in the model's predictive capabilities. Following this notable surge, both the training and cross-validation lines maintain a steady and consistent accuracy level until the conclusion of the learning curve.

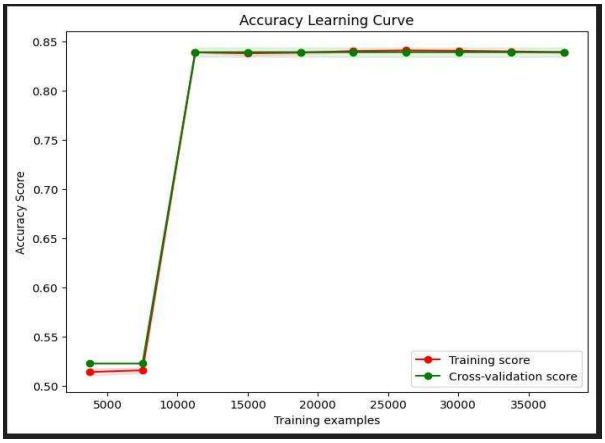

Fig. 4 Accuracy Learning Curve – Decision Tree (80:20)

The classification summary for the Decision Tree model provides a detailed evaluation of its performance in a binary classification task involving classes 0 and 1. The precision values indicate that the model achieved 85% accuracy in predicting instances of class 0 and 83% accuracy for class 1. The recall values demonstrate the model's ability to capture actual cases, with 80% recall for class 0 and 88% for class 1. The F1-scores, harmonizing precision and recall, are 0.82 for class 0 and 0.85 for class 1. The support values specify that the dataset has 5588 instances of class 0 and 6141 instances of class 1.

TABLE X
CLASSIFICATION SUMMARY – DECISION TREE (80:20)

Classification summary – Decision Tree (80:20)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.85	0.80	0.82	5588
1	0.83	0.88	0.85	6141
Accuracy			0.84	11729
Macro Avg	0.84	0.84	0.84	11729
Weighted Avg	0.84	0.84	0.84	11729

The model's overall accuracy is 84%, indicating correct predictions for approximately 84% of the total instances. The macro-average and weighted average for precision, recall, and F1-score are consistently 0.84, reflecting a well-rounded performance across both classes.

TABLE XI
CLASSIFICATION SUMMARY – DECISION TREE (70:30)

Classification summary – Decision Tree (70:30)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.85	0.80	0.83	8412
1	0.83	0.88	0.85	9182
Accuracy			0.84	17594
Macro Avg	0.84	0.84	0.84	17594
Weighted Avg	0.84	0.84	0.84	17594

The model's overall accuracy is reported at 84%, indicating accurate predictions for approximately 84% of the total instances. Both macro-average and weighted-average values for precision, recall, and F1-score consistently stand at 0.84, portraying a consistently well-rounded performance across both classes.

TABLE XII
CLASSIFICATION SUMMARY – DECISION TREE (90:10)

Classification summary – Decision Tree (90:10)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.79	0.90	0.84	2770
1	0.90	0.79	0.84	3095
Accuracy			0.84	5865
Macro Avg	0.85	0.84	0.84	5865
Weighted Avg	0.85	0.84	0.84	5865

The model's overall accuracy is 84%, indicating correct predictions for approximately 84% of the total instances. The macro-average and weighted average for precision, recall, and F1-score are consistently 0.84.

G. Extreme Boosting

The Extreme Gradient Boosting accuracy learning curve reveals different patterns for both the training and cross-validation score lines. The training scoreline starts on an extremely high note, with a fantastic accuracy level of 98%. This initial top, however, is followed by a slow drop, settling at 95.5% accuracy. The cross-validation score line, on the other hand, starts with a lower accuracy of 89.5% and rises throughout the length of the learning curve, achieving a high accuracy of 91.5%.

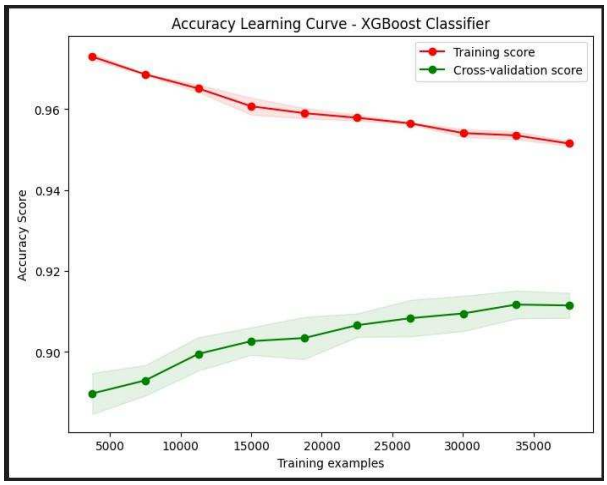

Fig. 5 Accuracy Learning Curve – Extreme Boost (80:20)

The classification summary for the Extreme Boosting model showcases its outstanding performance in a binary classification task with classes 0 and 1. The precision metrics reveal that the model achieved 92% accuracy in predicting instances of class 0 and 91% accuracy for class 1. Similarly, the recall metrics indicate the model's ability to capture actual cases, with 90% recall for class 0 and 93% for class 1.

TABLE XIII
CLASSIFICATION SUMMARY – EXTREME BOOST (80:20)

Classification summary – Extreme Boost (80:20)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.92	0.90	0.91	5588
1	0.91	0.93	0.92	6141
Accuracy			0.92	11729
Macro Avg	0.92	0.92	0.92	11729
Weighted Avg	0.92	0.92	0.92	11729

The model's overall accuracy is reported at an impressive 92%, signifying correct predictions for approximately 92% of the total instances. The macro-average and weighted average for precision, recall, and F1-score are consistently 0.92, affirming the model's reliability and robustness across both classes. The Extreme Boosting model demonstrates superior accuracy and effectiveness in accurately classifying instances from the provided test dataset.

TABLE XIV
CLASSIFICATION SUMMARY – EXTREME BOOST (70:30)

Classification summary – Extreme Boost (70:30)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.91	0.91	0.91	8412
1	0.91	0.92	0.92	9182
Accuracy			0.91	17594
Macro Avg	0.91	0.91	0.91	17594
Weighted Avg	0.91	0.91	0.91	17594

The model's overall accuracy is an impressive 91%, indicating accurate predictions for approximately 91% of the total instances. Both macro-average and weighted-average values for precision, recall, and F1-score consistently stand at 0.91, affirming the model's reliability and robustness across both classes.

TABLE XV
CLASSIFICATION SUMMARY – EXTREME BOOST (90:10)

Classification summary – Extreme Boost (90:10)				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.92	0.90	0.91	2770
1	0.91	0.93	0.92	3095
Accuracy			0.91	5865
Macro Avg	0.92	0.91	0.91	5865
Weighted Avg	0.91	0.91	0.91	5865

The model's overall accuracy is reported at an impressive 91%, signifying correct predictions for approximately 91% of the total instances. The macro-average and weighted average for precision, recall, and F1-score are consistently 0.92 and 0.91.

H. Comparison among All Models Result

Support Vector Machines, Random Forest, Naïve Bayes, Decision Tree, and Extreme Boosting. The metrics assessed include Accuracy, Precision, Recall, and F1-Score, each providing a unique perspective on the models' capabilities.

TABLE XVI
COMPARISON BETWEEN ALL MODELS RESULT

Comparison of All Models				
	Precision	**Recall**	**F1-Score**	**Support**
0	0.92	0.90	0.91	2770
SVM	82%	82%	82%	82%
Random Forest	85%	85%	85%	85%
Naïve Bayes	73%	78%	74%	72%
Decision Tree	84%	84%	84%	84%
Extreme Boosting	92%	92%	92%	92%

Extreme Boosting emerges as the top performer across the board, exhibiting an outstanding 92% accuracy, precision, recall, and F1 score. This exceptional consistency signifies the robust predictive capabilities of Extreme Boosting, making it a compelling choice for applications where high precision and recall are paramount. Random Forest follows closely, achieving an 85% accuracy and excelling in all other metrics with corresponding percentages. Naïve Bayes demonstrates a balanced performance with a 73% accuracy while showcasing commendable Precision, Recall, and F1-Score values. Support Vector Machines (SVM) and Decision Tree consistently perform, securing 82% accuracy.

I. Phishing Domain Detection System Prototype Testing

The text box on the left is for the user to paste the link that needs to be checked, and the output will be automatically generated in the left text box labelled “output”. If the link is detected as phishing, the output will display “Phishing” while “Not Phishing” if the link is detected as safe. The extreme Boosting model is applied to the system prototype because it outperforms the other four models in terms of overall performance.

The extreme Boosting model is applied to the system prototype because it outperforms the other four models’ overall performance. The testing process includes diverse datasets from Google, and the phishing site will be taken from PhishTank, a website that lists many malicious sites using a Denylist method. All the links on the site were posted by users who had already become victims of the links.

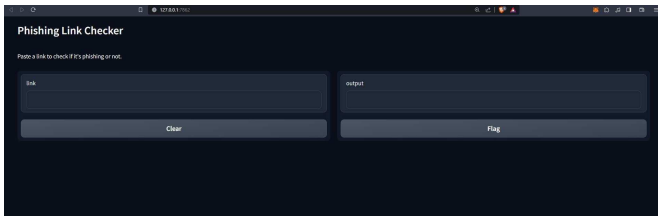

Fig. 6 User Interface of System Prototype

IV. Conclusion

This research primarily aimed to create a phishing domain identification model using machine learning techniques. By providing the results of tests done to assess the performance of machine learning algorithms, this study seeks to aid in developing reliable techniques for recognizing and blocking phishing domains. Additionally, this general aim was broken down into three goals.

The first objective was to apply machine learning algorithms to identify fraudulent domains associated with phishing activities. Many literature reviews, studies during preliminary research, and knowledge acquisition have been conducted to accomplish the objective. The primary focus of this project was to create machine learning models that can identify the traits and attributes that differentiate phishing attacks from legitimate domains. As outlined in the chapters, the efficacy of these models provides tangible evidence of achieving the second objective. The study showed a solid understanding of the complexities of identifying legitimate and malicious domains by employing methodologies such as Decision Tree, Naïve Bayes, Support Vector Machine (SVM), Random Forest, and Extreme Boosting. This prototype sought to provide users, particularly businesses, with a tool to safeguard themselves against phishing scams. The Extreme Boosting model accomplishes this objective applied to the system prototype as the other model outperforms the model's accuracy, and the system prototype can display the correct output of legitimate and phishing sites.

Several ideas and options for future improvement arise in light of the stated strengths and limitations. Implementing a mechanism for continuous model training is proposed to enhance the system's adaptability to the evolving landscape of phishing techniques. Regular updates based on new data can bolster the model's resilience against emerging threats. Additionally, diversifying the dataset's sources is recommended to ensure a more comprehensive representation of phishing scenarios, improving the model's ability to generalize to different attack vectors.

References

[1] C. Pascariu and I. C. Bacivarov, "Detecting phishing websites through domain and content analysis," in *Proc. 13th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, 2021. doi: 10.1109/ecai52376.2021.9515165.

[2] P. Bhatt, M. S. Obaidat, G. Dangwal, A. K. Das, M. Wazid, and B. Sadoun, "Machine learning-based security mechanism for detecting phishing attacks," in *2024 Int. Conf. Commun., Comput., Cybersecurity, Inform. (CCCI)*, Oct. 2024, pp. 1–6. doi: 10.1109/ccci61916.2024.10736460.

[3] M. Chatterjee and A. S. Namin, "Detecting phishing websites through deep reinforcement learning," in *Proc. Int. Comput. Softw. Appl. Conf. (COMPSAC)*, 2019. doi: 10.1109/compsac.2019.10211.

[4] S. Ahmad, M. A. Haque, H. A. M. Abdeljaber, M. U. Bokhari, J. Nazeer, and B. K. Mishra, "Phishing website detection: A dataset-centric approach for enhanced security," *Data Metadata*, vol. 3, Dec. 2024. doi: 10.56294/dm2024.223.

[5] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Gener. Comput. Syst.*, vol. 94, 2019. doi: 10.1016/j.future.2018.11.004.

[6] T. N. S. Charishma, A. S. Koushik, G. S. A. Reddy, and M. HimaBindu, "Employing machine learning algorithms to detect phishing URL websites," in *2024 Int. Conf. IoT Based Control Netw. Intell. Syst. (ICICNIS)*, Dec. 2024, pp. 1553–1558. doi: 10.1109/icicnis64247.2024.10823220.

[7] Y. H. Jazyah and L. Al Shalabi, "Phishing detection using clustering and machine learning," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 13, no. 4, p. 4526, Dec. 2024. doi: 10.11591/ijai.v13.i4.pp4526-4536.

[8] M. Amanullah, V. Selvakumar, A. Jyot, N. Purohit, S. Shitharth, and M. Fahlevi, "CNN-based prediction analysis for web phishing prevention," in *Int. Conf. Edge Comput. Appl. (ICECAA)*, 2022. doi: 10.1109/icecaa55415.2022.9936112.

[9] P. Y and U. Sree, "Phishing website detection using machine learning," *J. Innov. Technol.*, vol. 2024, no. 1, Nov. 2024. doi: 10.61453/joit.v2024no30.

[10] K. S. N. Sushma, M. Jayalakshmi, and T. Guha, "Deep learning for phishing website detection," in *MysuruCon 2022 - 2022 IEEE 2nd Mysore Sub Sect. Int. Conf.*, 2022. doi: 10.1109/mysurucon55714.2022.9972621.

[11] D. Zinca and A. Negrea, "Comparative study of phishing URL detection using artificial intelligence algorithms," in *2024 Int. Symp. Electron. Telecommun. (ISETC)*, Nov. 2024, pp. 1–4. doi: 10.1109/isetc63109.2024.10797218.

[12] R. Wazirali, R. Ahmad, and A. A. K. Abu-Ein, "Sustaining accurate detection of phishing URLs using SDN and feature selection approaches," *Comput. Netw.*, vol. 201, 2021. doi: 10.1016/j.comnet.2021.108591.

[13] M. El-Rashidy, "A smart model for web phishing detection based on new proposed feature selection technique," *Menoufia J. Electron. Eng. Res.*, vol. 0, no. 0, 2020. doi: 10.21608/mjeer.2020.32404.1021.

[14] N. B. M. Noh and M. N. B. M. Basri, "Phishing website detection using random forest and support vector machine: A comparison," in *2021 2nd Int. Conf. Artif. Intell. Data Sci. (AiDAS)*, 2021. doi: 10.1109/aidas53897.2021.9574282.

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

[16] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *Comput. Secur.*, vol. 104, 2021. doi: 10.1016/j.cose.2020.102123.

[17] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Knowl. Inf. Syst.*, vol. 64, no. 6, 2022. doi: 10.1007/s10115-022-01672-x.

[18] V. Borate, A. Adsul, R. Dhakane, S. Gawade, S. Ghodake, and P. Jadhav, "A comprehensive review of phishing attack detection using machine learning techniques," *Int. J. Adv. Res. Sci., Commun. Technol.*, pp. 435–441, Oct. 2024. doi: 10.48175/ijarsct-19963.

[19] P. Sharma, B. Dash, and M. F. Ansari, "Anti-phishing techniques – A review of cyber defense mechanisms," *IJARCCE*, vol. 11, no. 7, 2022. doi: 10.17148/ijarcce.2022.11728.

[20] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 2, 2022. doi: 10.1016/j.jksuci.2019.12.005.

[21] S. Alnemari and M. Alshammari, "Detecting phishing domains using machine learning," *Appl. Sci.*, vol. 13, no. 8, 2023. doi: 10.3390/app13084649.

[22] C. S. Shieh, W. W. Lin, T. T. Nguyen, C. H. Chen, M. F. Horng, and D. Miu, "Detection of unknown DDoS attacks with deep learning and Gaussian mixture model," *Appl. Sci.*, vol. 11, no. 11, 2021. doi: 10.3390/app11115213.

[23] R. Mahajan and I. Siddavatam, "Phishing website detection using machine learning algorithms," *Int. J. Comput. Appl.*, vol. 181, no. 23, 2018. doi: 10.5120/ijca2018918026.

[24] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, 2019. doi: 10.1016/j.eswa.2018.09.029.

[25] R. Amrish, K. Bavapriyan, V. Gopinaath, A. Jawahar, and C. V. Kumar, "DDoS detection using machine learning techniques," *J. ISMAC*, vol. 4, no. 1, 2022. doi: 10.36548/jismac.2022.1.003.

[26] S. P, K. S, R. K. S, G. P. G. R, P. M, and D. B, "Evaluating the efficacy of machine learning methods in phishing detection: A comparative analysis," in *2024 IEEE Int. Conf. Blockchain Distrib. Syst. Secur. (ICBDS)*, Oct. 2024, pp. 1–7. doi: 10.1109/icbds61829.2024.10837224.

[27] S. Merugula, K. S. Kumar, S. Muppidi, and C. Vidyadhari, "Stop phishing: Master anti-phishing techniques," in *2022 IEEE North Karnataka Subsection Flagship Int. Conf. (NKCon)*, 2022. doi: 10.1109/nkcon56289.2022.10126569.

[28] R. Tamilkodi, A. Harika, M. Harika, P. V. S. Abhilash, C. H. Sai, and P. Ps, "Enhanced security measures against phishing threats," in *2024 5th Int. Conf. Data Intell. Cogn. Inform. (ICDICI)*, Nov. 2024, pp. 62–67. doi: 10.1109/icdici62993.2024.10810772.

[29] "Enhanced phishing detection: An ensemble stacking model with DT-RFECV and SMOTE," *Appl. Math. Inf. Sci.*, vol. 18, no. 6, pp. 1481–1493, Nov. 2024. doi: 10.18576/amis/180624.

[30] A. Pathak, P. Pandey, and V. Raheja, "Comparing different machine learning techniques for detecting phishing websites," in *AI in the Social and Business World: A Comprehensive Approach*, Bentham Science Publishers, 2024, pp. 222–234. doi: 10.2174/9789815256864124010012.