*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 2-B, pp. 569–599
DOI : 10.47974/JDMSC-1893

# Discrete mathematical models for enhancing cybersecurity : A mathematical and statistical analysis of machine learning approaches in phishing attack detection

Dinesh Goyal [‡]
*Department of Computer Science & Engineering*
*Poornima Institute of Engineering & Technology*
*Jaipur*
*Rajasthan*
*India*

Farhan Sheth [†]
*Department of Computer Science & Engineering*
*Manipal University Jaipur*
*Jaipur*
*Rajasthan*
*India*

Priya Mathur [§]
*Department of Mathematics*
*Poornima Institute of Engineering & Technology*
*Jaipur*
*Rajasthan*
*India*

Amit Kumar Gupta *
*Department of Computer Science & Engineering*
*Manipal University Jaipur*
*Jaipur*
*Rajasthan*
*India*

[‡] *E-mail:* dinesh.goyal@poornima.org
[†] *E-mail:* farhansheth.jb@gmail.com
[§] *E-mail:* drpriyamathur21@gmail.com
* *E-mail:* dramitkumargupta1983@gmail.com (Corresponding Author)

**Abstract**

This paper presents a discreate mathematical modelling of cybersecurity phishing attack detection methodologies, emphasizing the crucial role of continual advancements in detection methods amidst the pervasive threat of phishing attacks in the cybersecurity landscape. Leveraging mathematical modeling and machine learning algorithms, the study employs three distinct datasets—Mendeley, URL tokenized, and a merged dataset integrating both. Multiple machine learning algorithms, including Logistic Regression, k-Nearest Neighbors, Support Vector Machines, Random Forest, Gradient Boosting Machines, Neural Networks, CatBoost, and XGBoost, are systematically applied to evaluate their efficacy. In the original Mendeley dataset, XGBoost achieves a top accuracy of 97.24%, along with CatBoost and Random Forest exceeding 97%. Post-preprocessing, CatBoost leads with an accuracy of 97.28%, showcasing superior precision, sensitivity, and F-score. Despite slight accuracy reductions post-preprocessing, models consistently achieve over 94% accuracy on the preprocessed Mendeley dataset, highlighting the substantial impact of preprocessing. Tokenized URLs exhibit comparatively lower performance, with the highest accuracy at 91.95%, emphasizing the challenges associated with this approach. The combined dataset proves optimal for most models, with XGBoost and SVM achieving the highest overall accuracy at 97.68%. SVM excels in sensitivity and specificity, while XGBoost excels in precision. The merged dataset significantly enhances accuracy, sensitivity, specificity, and precision, underscoring its pivotal role in refining predictive capabilities for identifying phishing websites. The results section provides a detailed overview of machine learning model performance on different datasets. CatBoost emerges as a standout performer on the preprocessed Mendeley dataset. The tokenized URLs offer valuable insights into associated challenges, and the combined dataset proves effective for various models. Confusion matrices, ROC curves, and Precision-Recall curves provide nuanced perspectives on model behavior, emphasizing the need for ongoing refinement and investigation into misclassification patterns to enhance model effectiveness in combating phishing threats.

***Subject Classification:*** *68T07, 68M25.*

***Keywords:*** *Phishing detection, Discrete mathematical modelling, Cybersecurity, Security algorithms, Security analytics.*

## 1. Introduction

Phishing attacks, posing a persistent threat in the ever-evolving cybersecurity landscape, underscore the critical need for continuous advancements in detection methods. This literature review conducts a comprehensive investigation into recent research endeavours, with a specific focus on the application of machine learning in the context of phishing detection. The foundational works of Gandotra and Gupta (2021), Omar et al. (2023), and other significant contributions set the stage for our exploration. Recognizing the dynamic and adaptive nature of phishing attacks, our research methodology is meticulously crafted to address the

multifaceted challenges associated with detection. We employ a diverse array of machine learning models, ranging from traditional approaches to state-of-the-art techniques. The inclusion of extensive pre-processing steps ensures that raw data is transformed and optimized for effective model training. Additionally, feature selection techniques are systematically applied to navigate the complex landscape of features, enhancing the models' ability to discern phishing patterns. The results of our study offer valuable insights into the efficacy of machine learning models when applied to distinct datasets. Through rigorous experimentation, we uncover the nuanced influence of various dataset characteristics on the overall accuracy of phishing detection models. This analysis sheds light on the adaptability and robustness of different methodologies in diverse scenarios, emphasizing the importance of tailoring detection strategies to specific dataset profiles. As the digital landscape undergoes continuous evolution, the imperative to comprehend and enhance phishing detection methodologies becomes increasingly apparent. The insights garnered from this research contribute to the ongoing discourse in the field of cybersecurity, providing a foundation for the development of more resilient and sophisticated tools to combat phishing threats. In essence, our work emphasizes the dynamic nature of cybersecurity challenges and underscores the necessity for innovative and adaptive approaches to ensure the resilience of digital ecosystems.

## 2. Related Study

The Gandotra and Gupta propose an efficient approach for phishing detection, emphasizing the role of machine learning. Their work delves into algorithm development and analysis, showcasing the importance of feature selection for improved accuracy [1]. The study by Omar et al. explores phishing behavior analysis and feature selection to enhance prediction rates. This paper contributes to understanding the intricacies of phishing behavior and the significance of relevant features in detection models [2]. Uddin et al. conduct a comparative analysis of ML-based website phishing detection, focusing on URL information. The research provides insights into the effectiveness of different machine learning approaches in a real-world context [3]. Ojewumi et al. evaluate the performance of various machine learning tools for detecting phishing attacks on web pages. Their work emphasizes the need for robust evaluation metrics and benchmarks in assessing the efficacy of detection models [4]. Jain and Gupta present a comparative analysis of feature-based

machine learning approaches for phishing detection. This study contributes valuable insights into the selection of features critical for distinguishing phishing websites [5]. Niakanlahiji et al. introduce Phishmon, a machine learning framework for detecting phishing webpages. Their work focuses on the development of a comprehensive framework for phishing detection, contributing to the evolving landscape of ML-based solutions [6]. Deshpande et al. propose a detection mechanism for phishing websites using machine learning. The paper contributes to the exploration of novel features and methodologies for enhancing the accuracy of phishing detection [7]. Almseidin et al. explore phishing detection based on machine learning and feature selection methods. Their work contributes to the understanding of the impact of feature selection on the overall performance of detection models [8]. Li et al. present a stacking model utilizing URL and HTML features for phishing webpage detection. The study emphasizes the integration of diverse features to improve the robustness of phishing detection systems [9]. Hannousse and Yahiouche focus on benchmark datasets for machine learning-based website phishing detection. Their work addresses the crucial need for standardized datasets to facilitate fair comparisons and evaluations of different detection models [10]. Mohamed et al. propose an effective and secure mechanism for phishing attacks using a machine learning approach. Their work contributes to the ongoing efforts to develop mechanisms that not only detect but also secure against phishing attacks [11]. Abdelhamid et al. conduct a comprehensive comparison of intelligent machine learning models based on content and features. The study highlights the importance of considering both content and features for an inclusive evaluation of phishing [13].

The literature reviewed demonstrates the diversity of approaches in the field of machine learning-based phishing detection. Key considerations include feature selection, benchmark datasets, and the integration of multiple features for enhanced accuracy. However, challenges such as evolving phishing tactics and the need for real-time detection remain areas for future research [14-17].

## 3. Methods and Material

The methodology is divided into two clear phases. The initial phase focuses on detailed data pre-processing, a crucial preparatory stage laying the groundwork for subsequent analyses. Following this preparation, the latter part of our approach employs a varied range of Machine Learning

algorithms and neural networks for classification. Algorithm 1 succinctly represents the study's algorithmic structure to elucidate the procedural details. The subsequent sections delve into each step's specifics, providing insights into their distinct contributions and performance nuances within the scope of this research.

**Algorithm 1. Methodology of the study**

**Input:** Dataset

Step 1. Data Acquisition:

- Input collected dataset for the analysis.

Step 2. Tokenization (NLP):

- Utilize Natural Language Processing (NLP) techniques for URL tokenization.
- Transform the URLs into a structured format suitable for subsequent analysis.

Step 3. Feature Selection:

- Employ the KBest method in conjunction with chi squared (Chi2) statistical test for feature selection.
- Identify and retain the most relevant features based on their statistical significance for the dataset and for tokens created from URL

Step 4. Data Splitting:

- Partition the preprocessed data into training and testing sets.

Step 5. Optimizing Features:

- Optimize the feature values to ensure uniformity and mitigate the impact of scale variations.
- Apply techniques such as Normalizer for tokenized data and subsequently applying Quantile Transformer to enhance algorithm convergence.

Step 6. Application of Machine Learning Algorithms:

- Implement a diverse set of Machine Learning algorithms for classification.

Step 7. Model Training and Evaluation:

- Train machine learning models on the training dataset.

- Evaluate the performance of each model on the testing dataset, employing relevant metrics such as accuracy, precision, recall, and F1-score.

Step 8. Results Analysis:

- Analyze the results obtained from different algorithms using confusion matrixes and ROC curves.
- Summarize the findings and draw conclusions regarding the efficacy of the proposed methodology.

**Output:**

Classification Results (Phishing or Legitimate URL)

### 3.1 *Dataset Characteristics*

The dataset employed in this research was obtained from the Mendeley Data repository [18]. Constructing a benchmark dataset for phishing website detection poses a significant challenge due to the short lifespan of such websites and the difficulty in generating synthetic URLs with distinctive features. Past studies have faced performance degradation when dealing with evolving datasets, even after retraining [2]. In this study, we have developed a robust detection model capable of classifying websites with higher accuracy. This is achieved by incorporating both URL details and the URLs themselves, transformed into tokens, for classification. This innovative approach holds the potential to yield superior results, effectively addressing the complexities associated with the dynamic nature of phishing websites. The model's adaptability enables seamless application in diverse scenarios. The improved classification accuracy and generalizability may enhance its utility across various applications within the cybersecurity domain.

#### 3.1.1 Dataset

The dataset utilized in this study encompasses a total of 11,430 URLs, exhibiting a balanced distribution with precisely 5,715 URLs assigned to each of the two classes: phishing and legitimate. Each URL, along with its corresponding classification, is associated with 87 distinct features. These features are systematically organized into three groups, each providing unique insights into the characterization of URLs. The first category encompasses 56 features derived from the structure and syntax of URLs shown in Table 1. These features offer a detailed examination of key elements such as URL length, composition, and various syntactic

components, providing a comprehensive perspective on the structural aspects of the URLs. The second category consists of 24 features dedicated to capturing attributes derived from the content of the linked pages shown in Table 2. These features delve into the semantic and contextual aspects of the URLs, encompassing factors such as phishing hints, the presence of login forms, and content-related metrics. This category significantly contributes to understanding the inherent nature of the linked web pages. The third category includes 7 features obtained through querying external services, adding an additional layer of depth to the dataset shown in Table 3. Metrics such as domain registration length, age, and various parameters obtained from external services enrich the dataset by incorporating valuable information obtained through systematic queries. This systematic categorization establishes a foundational framework for subsequent analyses, offering a structured approach to interpreting these features within the dataset [3]. The features within each category collectively contribute to a comprehensive understanding of the URLs, facilitating a nuanced exploration of their structural, content-related, and external characteristics. Where nb stands for number of. The dataset obtained from Mendeley Data repository, has been referred as Mendeley dataset in the study.

**Table 1**

**Features from the structure and syntax of URLs**

| | | | | |
|---|---|---|---|---|
| length_url | length_hostname | ip | nb_dots | nb_hyphens |
| nb_at | nb_qm | nb_and | nb_or | nb_eq |
| nb_underscore | nb_tilde | nb_percent | nb_slash | nb_star |
| nb_colon | nb_comma | nb_ semicolumn | nb_dollar | nb_space |
| nb_www | nb_com | nb_dslash | http_in_ path | https_token |
| ratio_digits_url | ratio_digits_host | punycode | port | tld_in_path |
| tld_in_subdomain | abnormal_ subdomain | nb_ subdomains | prefix_ suffix | random_ domain |
| shortening_ service | path_extension | nb_ redirection | nb_ external_ redirection | length_ words_raw |
| char_repeat | shortest_words_ raw | shortest_ word_host | shortest_ word_path | longest_ words_raw |

*Contd...*

| | | | | |
|---|---|---|---|---|
| longest_word_ host | longest_word_ path | avg_words_ raw | avg_word_ host | avg_word_ path |
| phish_hints | domain_in_brand | brand_in_ subdomain | brand_in_ path | suspicious_ tld |
| statistical_report | | | | |

**Table 2**

**Features from the content of their corresponding pages**

| | | | | |
|---|---|---|---|---|
| external_ favicon | nb_hyperlinks | ratio_ intHyperlinks | ratio_ extHyperlinks | ratio_ nullHyperlinks |
| nb_extCSS | ratio_ intRedirection | ratio_ extRedirection | ratio_intErrors | ratio_extErrors |
| login_form | links_in_tags | submit_email | ratio_intMedia | ratio_extMedia |
| sfh | iframe | popup_window | safe_anchor | onmouseover |
| right_clic | empty_title | domain_in_title | domain_with_ copyright | |

**Table 3**

**Features from querying other services**

| | | | | |
|---|---|---|---|---|
| page_rank | google_index | dns_ record | web_ traffic | domain age |
| domain_registration_ length | whois_registered_ domain | | | |

3.1.2 Features

By leveraging the CatBoost machine learning model, we generated a comprehensive visual representation showcasing the 30 most influential features obtained through the application of mutual information classification (mutual_info_classif) [19], as depicted in Figure 1. Upon examination of the image, it becomes apparent that, aside from the URL feature, the remaining top five features that significantly contribute to the classification task include page rank, Google index, the number of hyperlinks, domain age, and the count of occurrences of "www." Although the importance of these features may vary with different models, this analysis establishes a foundational understanding of the primary

contributors to the classification task. In Figure 2, we present the correlation matrix of the top features, providing valuable insights into the relationships among distinctive features. Notably, the observation reveals a high correlation between "nb_eq" and "nb_and," as well as between "links_in_ tags" and "ratio_intHyperlinks." Furthermore, a strong correlation is observed between "ratio_digits_url" and "IP." This correlation matrix serves as a crucial tool for discerning interdependencies among features, thereby enhancing our comprehension of the intricate relationships within the dataset. The visual representation and correlation matrix collectively contribute to a more nuanced understanding of feature importance and relationships, aiding in the interpretability and optimization of the machine learning model.

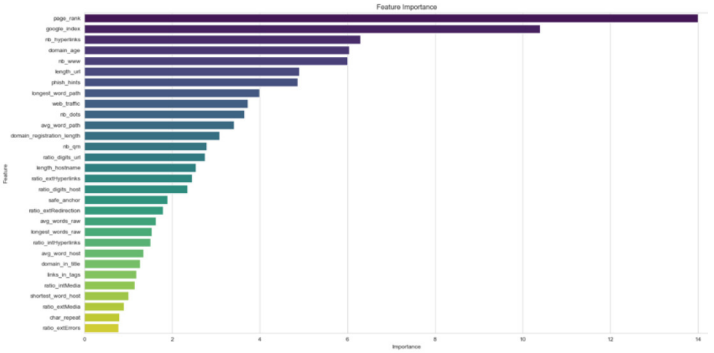



**Figure 1**

**Important Features Map for Classification**

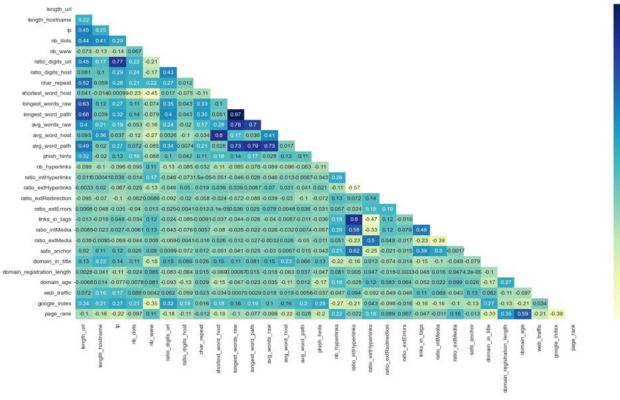


**Figure 2**

**Correlation matrix plot of important features**

### 3.2 *Preprocessing*

Preprocessing stands as a pivotal step preceding model training, playing a vital role in readying datasets for effective model training. This multifaceted phase involves essential tasks such as data cleaning and standardization, ensuring that the input data is refined and optimized for subsequent modeling processes. By undertaking these preprocessing steps, the dataset is groomed to enhance generalization and predictive capabilities during the training phase, contributing to the overall robustness and performance of the machine learning model.

#### 3.2.1 Natural Language Processing

The URL holds intrinsic value, serving as a pivotal resource for assessing the authenticity of a website by scrutinizing various attributes such as the length and format of the URL, as evidenced by insights derived from the correlation matrix. Employing Natural Language Processing (NLP) techniques, the URL undergoes body text analysis. This involves tokenization, where the URL is deconstructed into individual words, referred to as tokens, to establish the meaning and context of the text by analyzing the sequences of the words. As a vital augmentation to the dataset, a new feature is introduced tokens extracted from the textual content [3][20]. The resulting frequency distribution of the number of tokens, categorized by class, is visually depicted in Figure 3. Following tokenization, the text undergoes a process of stemming, applying linguistic normalization to reduce words to their root form. This step ensures a more coherent and standardized representation of the textual content [21]. The stemmed text is then incorporated into a Count Vectorizer to convert the processed text into a numerical format suitable for machine learning algorithms. The Count Vectorizer functions by creating a sparse matrix that represents the frequency of each stemmed word in the text. This transformation enhances the efficiency of subsequent machine learning models by converting the textual data into a structured and quantifiable format. The sparse matrix generated by the Count Vectorizer is further transfigured into a structured dataframe, optimizing the dataset for ease of training. This conversion not only enhances the interpretability and accessibility of the features but also ensures that the dataset is well-prepared for subsequent machine learning tasks.

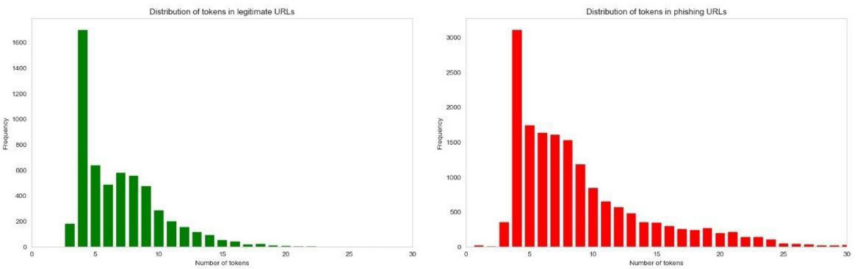



**Figure 3**

**Number of Tokens vs Frequency for legitimate and Phishing URLs**

3.2.2 Feature Selection

Feature selection emerges as a crucial step in the preprocessing phase, wielding a profound impact on the model's accuracy. The careful selection of features has the potential to significantly enhance model performance, while the inclusion of unsuitable features can detrimentally affect outcomes. This critical process involves extracting extensive array of features, subjecting them to statistical scrutiny, and subsequently prioritizing the most weighted and crucial aspects. The outcome is a set of dataset-dependent features that optimally contribute to superior model performance. For both the Mendeley dataset and the URL tokenized dataset, feature selection is executed employing KBest and Chi2 methodologies. The Chi2 algorithm leverages the $\chi^2$ statistic, using it iteratively to discretize numeric attributes until inconsistencies emerge within the data. This two-phase process ensures effective feature selection through discretization [22]. The $\chi^2$ statistic is computed using Equation (1), providing a quantitative measure for the significance of each feature. This approach guarantees that the selected features align with the dataset characteristics, optimizing the model's predictive capabilities. The judicious application of feature selection methodologies is essential in tailoring the dataset to the model's specific requirements, ultimately improving the model's ability to make accurate predictions.

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{n}\frac{(P_{ij} - Q_{ij})^2}{Q_{ij}} \tag{1}$$

Where $n$ is the number of classes, $P_{ij}$ is the number of patterns in the $i^{th}$ interval of $j^{th}$ class and $Q_{ij}$ is the expected frequency of $P_{ij}$. Then using the

KBest, which selects the features according to the $k$ highest scores, the features are selected.

### 3.2.3 Optimizing Features

Dealing with irregularities and variations in numerical data poses a significant challenge, directly impacting the accuracy of machine learning models. Addressing this challenge is crucial, and one key approach involves optimizing features to contribute to the enhancement of model accuracy and overall performance. In the case of the URL tokenized dataset, optimization is achieved through normalization. This process involves scaling individual samples to achieve unit norm, where each sample with at least one non-zero component is independently rescaled. The goal is to standardize the norm of each sample to equal one. Moreover, both the Mendeley dataset and the merged dataset, which combines the Mendeley dataset and the URL tokenized dataset, undergo standardization using the Quantile Transformer, a non-linear transformation. This transformation is characterized by its monotonic nature, preserving the rank of values along each feature. By applying this technique, the merged dataset aligns all its features into the same desired distribution based on Equation (2). This standardization process ensures consistency and comparability across features, contributing to a more robust and reliable modeling environment.

$$G^{-1}(F(X)) \tag{2}$$

Where $G^{-1}$ is the quantile function of the desired output distribution $G$ and $F$ is the cumulative distribution function of the feature. By performing a rank transformation, the quantile transforms smooth out unusual distributions and is less influenced by outliers. This standardization process facilitates a consistent and comparable representation of features, contributing to the robustness and reliability of subsequent analyses and model training.

### 3.3 *Machine Learning Algorithms*

Machine learning, a dynamic field, is defined as the study that empowers computers to learn without explicit programming. It's remarkable ascent in recent years has led to a proliferation of applications across various domains, including but not limited to robotics, autonomous vehicle control, speech processing, natural language processing, neuroscience research, and computer vision. One prominent aspect of

machine learning is supervised learning, which focuses on mapping task input to output. This entails training algorithms on labeled datasets, enabling them to discern patterns and subsequently apply acquired knowledge to predict or classify unseen data in a test dataset [23]. In the context of cybersecurity, the application of machine learning is particularly noteworthy. Network logs can reveal tracks of legal or illegal activity, and to differentiate between them, machine learning algorithms are applied. Machine learning algorithms are also successfully employed as collaborative classifiers for intrusion detection [24]. In the domain of identifying phishing and legitimate websites, this study has utilized eight distinct machine learning models. The effectiveness of machine learning models in detecting phishing attempts relies on their ability to analyze and generalize from a diverse set of training examples [25]. The strategic use of diverse machine learning models enhances the accuracy and robustness of detection mechanisms, contributing significantly to ongoing efforts to mitigate cyber threats in an ever-evolving digital landscape.

### 3.3.1 Logistic Regression

Logistic Regression models, categorized as statistical models, prove to be valuable tools for elucidating the relationship between a qualitative dependent variable with discrete values and one or more independent variables. Precisely, these models assert that the logarithm of the odds of the dependent variable (Y) being equal to 1 is linearly associated with the predictor variable(s). This relationship is expressed mathematically in Equation (3) [26], where the odds represent the likelihood of the event (Y = 1) occurring based on the values of the predictor variable X.

$$log\ log\ odds[Y = 1 \mid X] = \beta_0 + \beta_1 X \tag{3}$$

In Equation (3), $\beta_0$ represents the intercept, and $\beta_1$ is the regression coefficient of X. Logistic regression finds widespread application in diverse fields, ranging from medical research to marketing analytics. Its utility is rooted in its ability to model the probability of a binary outcome, making it a robust tool for scenarios where the dependent variable takes on discrete values. The model's flexibility and interpretability contribute to its broad usage in various domains, allowing for effective analysis and prediction of binary outcomes in real-world applications.

### 3.3.2 K-Nearest Neighbors (kNN)

The k-Nearest Neighbors (kNN) algorithm, classified as an instance-based or non-generalizing learning classification method, operates on a

straightforward principle. It preserves all existing cases and classifies new instances by employing a similarity measure, typically utilizing distance functions such as Euclidean (Equation (4)) or Manhattan (Equation (5)) for continuous variables and Hamming Distance for categorical variables. The assignment of a class for a given value depends on the class that is most prevalent among its K nearest neighbors, as determined by the specified distance function employed [27].

$$D = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{4}$$

$$D = \sum_{i=1}^{k} \left| x_i - y_i \right| \tag{5}$$

The kNN algorithm exemplifies an approach where the classification of an instance is influenced by its proximity to other instances in the feature space. By utilizing distance metrics, it evaluates the similarity between the new instance and existing cases, assigning the class label based on the majority class within the K nearest neighbors. This characteristic renders kNN particularly adaptable to various types of data, whether continuous or categorical, highlighting its utility in diverse domains of machine learning and pattern recognition. The method's flexibility and simplicity contribute to its effectiveness in scenarios where the underlying data distribution may be complex or lacks a clear parametric form.

### 3.3.3 Support Vector Machine (SVM)

The Support Vector Machine (SVM) stands as a comprehensive class of machine learning models, finding widespread applications across various domains. Its inherent appeal lies in its ability to proficiently handle both classification and regression tasks. By leveraging kernel functions, SVM excels in navigating high-dimensional spaces, facilitating the effective modeling of non-linear relationships within the data [28]. The success of SVM can be attributed to its distinctive approach in defining decision boundaries. SVM aims to discover a hyperplane that maximizes the separation between different classes within the feature space. This strategically positioned hyperplane is crucial for maintaining the maximum margin between classes, thereby enhancing the model's ability to generalize. This meticulous process ensures that the chosen hyperplane not only accurately classifies the training data but also generalizes

effectively to unseen instances [29]. The SVM classification function, denoted as F (x), is a fundamental component of its operation. It serves as the mechanism through which SVM assigns new instances to different classes based on learned patterns from the training data. The formulation of the SVM classification function is expressed by Equation (6)[30]:

$$F(x) = w \cdot x - b \tag{6}$$

Where $w$ is the weight vector, $b$ is the bias and $x$ is a $n$-dimensional real vector. The unique approach to constructing decision boundaries, coupled with the rigorous mathematics embedded in its formula, underlines SVM's significance in contemporary machine learning applications.

### 3.3.4 Random Forest

The Random Forest model stands out as an exceptional entity within the realm of machine learning, offering a seamless blend of prediction accuracy and model interpretability. Its distinctive strengths arise from the strategic amalgamation of random sampling and ensemble strategies, enabling accurate predictions while concurrently enhancing generalization. This notable generalization property is a direct outcome of the bagging scheme, a mechanism designed to mitigate variance. Essentially, Random Forests inherit the advantages of decision trees but elevate their performance by incorporating bagging on samples, employing random subsets of variables, and adopting a majority voting scheme [31]. A distinctive feature that sets Random Forests apart from classical decision trees is their innate ability to sidestep the necessity for tree pruning. This is attributed to the ensemble and bootstrapping schemes, which collaboratively navigate and address overfitting issues. Within the Random Forest paradigm, each tree in the ensemble is meticulously constructed from a sample drawn with replacement from the training set, ensuring robustness and diversity in the model. During the node-splitting phase of tree construction, the best split is determined from all input features, introducing an additional layer of randomness. These intentional sources of randomness are strategically harnessed to diminish the variance of the forest estimator. In contrast to individual decision trees, notorious for their high variance and propensity to overfit, Random Forests succeed in curbing variance by amalgamating diverse trees, albeit with a marginal trade-off of a slight increase in bias. The variance reduction achieved by Random Forests is often substantial, resulting in an overall superior

model. This nuanced approach not only enhances predictive accuracy but also renders the model more interpretable, making Random Forest a robust and versatile choice in various machine learning applications [31].

$$F(x) = \left( sign(\sum_{i=1}^{N} w_i I(h_i(x) = j \right) \quad (7)$$

Here, N is the number of trees in the ensemble, $w_i$ are the weights assigned to each tree, $h_i(x)$ represents the prediction of the $i^{th}$ tree, and j denotes the class label. The classification function combines the predictions of multiple trees, resulting in an aggregated decision that contributes to the overall Random Forest prediction.

### 3.3.5 Gradient Boosting

Gradient Boosting Machines (GBMs) stand as a family of machine-learning techniques that have demonstrated remarkable success across a diverse array of practical applications. Their adaptability is a hallmark of their efficacy, allowing customization to the specific requirements of each application, including the flexibility to learn with respect to different loss functions. Commonly referred to as GBMs, these machines employ a learning procedure that sequentially fits new models, progressively refining their accuracy in estimating the response variable. The fundamental concept underpinning the algorithm involves constructing new base-learners that are optimally correlated with the negative gradient of the loss function associated with the entire ensemble. This strategic alignment ensures that each subsequent model complements and enhances the overall predictive power of the ensemble **[32]**.

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x) \quad (8)$$

Here, $F_m(x)$ represents the mm-th model in the ensemble, $F_{m-1}(x)$ is the aggregated prediction from the previous m−1 models, $\gamma$ is the learning rate (a small positive value that controls the contribution of each new model), and $h_m(x)$ is the new base-learner being added to the ensemble at the $m^{th}$ iteration. The algorithm sequentially refines the model by fitting new base-learners, optimizing their correlation with the negative gradient of the loss function.

### 3.3.6 CatBoost

CatBoost, a gradient boosting toolkit, distinguishes itself by introducing ordered boosting, a permutation-driven alternative to

conventional algorithms. It also incorporates an innovative algorithm specifically designed for the efficient processing of categorical features. Built upon the foundation of gradient-boosted decision trees, CatBoost's standout feature lies in its specialized algorithm, crafted to handle categorical features seamlessly, thus mitigating the need for extensive hyperparameter tuning [33].

This characteristic not only expedites the model development process but also reduces the likelihood of overfitting, fostering the creation of more generalized models.

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{9}$$

Here, F(x) represents the overall prediction, $\gamma_m$ is the learning rate for the $m^{th}$ iteration, and $h_m(x)$ is the contribution from the $m^{th}$ decision tree in the ensemble. The CatBoost algorithm sequentially adds decision trees, optimizing their contribution to the overall prediction while efficiently handling categorical features.

### 3.3.7 XGBoost

XGBoost stands out as a scalable end-to-end tree boosting system, featuring a novel sparsity-aware algorithm designed for sparse data and leveraging a weighted quantile sketch for approximate tree learning. The key driver behind the success of XGBoost lies in its scalability across a spectrum of scenarios, showcasing robust performance. What sets XGBoost apart is its theoretically justified weighted quantile sketch procedure, allowing the incorporation of instance weights in the process of approximate tree learning [34]. This consideration for instance weights enhances the model's adaptability to scenarios where certain data points carry more significance, contributing to a more nuanced and accurate representation of the underlying patterns in the data. The distributed nature of XGBoost ensures efficient utilization of computational resources, making it well-suited for large-scale datasets and distributed computing environments. This scalability, coupled with its flexibility and portability, positions XGBoost as a robust, efficient, and adaptable tool for machine learning.

$$F(x) = \sum_{k=1}^{K} f_k(x), \; f_k \in F \tag{10}$$

Here, F(x) represents the overall prediction, $\sum_{k=1}^{K} f_k(x)$ is the sum of individual tree predictions, and $f_k \in F$ denotes the space of possible trees. The weighted quantile sketch procedure in XGBoost plays a pivotal role in optimizing the learning process, particularly when dealing with sparse data and accommodating instance weights.

3.3.8 Neural Networks

The multilayer perceptron (MLP) stands as the most widely recognized and frequently employed type of neural network. In its typical configuration, signal transmission occurs unidirectionally within the network, flowing from input to output. This architecture, known as feedforward, is characterized by the absence of loops, ensuring that the output of each neuron does not influence the neuron itself. A key attribute enhancing the efficacy of the multilayer perceptron is the incorporation of non-linear activation functions. Presently, the most prevalent choices include the single-pole (or logistic) sigmoid, as defined in Equation (11), and the bipolar sigmoid, commonly known as the hyperbolic tangent, presented in Equation (12). These activation functions play a pivotal role in introducing non-linearity to the model, enabling it to capture intricate patterns and relationships within the data [34]. The learning process of the multilayer perceptron hinges on the minimization of measurement errors between the network's outputs and the desired outputs. This necessitates a backpropagation mechanism through the network, resembling the learning process itself. The overarching objective is to iteratively refine the connection weights by identifying the minimum of the error function. This approach is integral to the optimization of the neural network's performance [34].

$$f(s) = \frac{1}{1+e^{-s}} \tag{11}$$

$$f(s) = \frac{1-e^{-a \cdot s}}{1+e^{-a \cdot s}} \tag{12}$$

3.4 *Evaluation*

3.4.1 Optimal Hyperparameters

Parameters play a pivotal role in determining the performance of a model. Adjusting these parameters is imperative to achieve optimal model outcomes while mitigating the risks of overfitting or underfitting,

ultimately contributing to higher accuracy [35]. To fine-tune these parameters, the technique of grid search was employed. Grid search systematically evaluates various combinations of hyperparameter values defined within a set. While this method may seem exhaustive, its effectiveness lies in its thorough exploration of a spectrum of possibilities to pinpoint the configuration that optimizes model performance. Through this meticulous approach, the optimal set of parameters is identified, enhancing the model's predictive capabilities. To streamline the process and avoid unnecessary computations, the models were initially trained using default settings. Subsequently, only the high-performing models were selected for further parameter optimization. Applying grid search exclusively to these select models is a strategic decision aimed at maximizing efficiency. This targeted approach minimizes computational overhead, ensuring that grid search is employed where its impact is most significant, thereby enhancing the performance of the chosen models.

### 3.4.1 Performance Metrics

The evaluation of model performance constitutes a crucial aspect of this study, and various metrics have been employed to gain comprehensive insights. The models undergo rigorous testing using a diverse set of evaluation metrics, including but not limited to Accuracy, Precision, Sensitivity, Specificity, F-score, and Area Under the Curve (AUC). These metrics are defined by the following equations:

$$\text{Accuracy} = \frac{\text{number of correct prediction}}{\text{total number of samples}} \tag{13}$$

$$\text{Precision} = \frac{TP}{TN + FP} \tag{14}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{16}$$

$$F - score = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{17}$$

The terms TP, TN, FP, and FN correspond to true positives, true negatives, false positives, and false negatives, respectively. With a balanced

class distribution in the dataset, accuracy emerges as a straightforward and effective metric for comparing the performance of various models. Additionally, detailed insights into model performance were obtained through the generation of a confusion matrix, ROC curve, and Precision-Recall Curve for all models. The confusion matrix provides a granular breakdown of model predictions, showcasing the interplay between true and false classifications. The ROC curve illustrates the trade-off between sensitivity and specificity across different thresholds, offering a comprehensive view of a model's discriminative power. Simultaneously, the Precision-Recall Curve delineates the precision-sensitivity relationship, providing a nuanced understanding of a model's performance, especially in scenarios where class imbalances exist. By incorporating these complementary evaluation tools, the study aims to provide a holistic and nuanced assessment of the models, capturing their performance characteristics from various perspectives. This multifaceted approach ensures a comprehensive understanding of the models' effectiveness in phishing detection.

## 4. Results and Discussion

The Mendeley Dataset consists of 11,430 URLs, evenly distributed between legitimate and phishing instances, totaling 5,715 URLs for each class. The dataset was strategically partitioned into two subsets: an 80% training set comprising 9,144 URLs and a 20% test set containing 2,286 URLs for all variations. The application of machine learning models produced results detailed in Table 3. Feature selection techniques, specifically KBest and Chi2, were applied to the Mendeley dataset to identify the most prominent features. Subsequently, the dataset was divided, and the identified prominent features underwent transformation and optimization using Quantile Transformer, culminating in the outcomes presented in Table 4. The URLs in the Mendeley dataset underwent preprocessing steps, including tokenization, stemming, and the utilization of count vectorization techniques, converting the collection of text documents into a matrix of token counts. These steps contributed to a refined representation of the URL data, enhancing the effectiveness of subsequent analysis. The classification results, post-normalization, are presented in Table 5. In addressing the challenge posed by the substantial feature space resulting from count vectorization, the preprocessed URL tokenized dataset underwent strategic refinement. Employing KBest and Chi2 feature selection methods, the most salient features were successfully

extracted, reducing the feature space to a more manageable scale. This was essential as the initial count vectorization had generated over 18,000 feature categories. Following this feature extraction, a synergistic approach was adopted to enhance model training. The preprocessed Mendeley dataset and the refined URL tokenized dataset were merged, aiming to harness the combined strengths of both the URL prominent features and the tokenized URL representations, fostering a more comprehensive and effective model training paradigm. The performance metrics observed on the test dataset are displayed in Table 6. This table provides a comprehensive overview of the models' efficacy in handling the merged datasets, shedding light on their performance in the context of the refined feature set and dual representation approach.

**Table 3**

**Mendeley Dataset classification performance**

| **Models** | **Accuracy (%)** | **Precision (%)** | **Sensitivity (%)** | **Specificity (%)** | **F-score (%)** | **AUC** |
|---|---|---|---|---|---|---|
| Logistic Regression | 78.78 | 78.19 | 79.09 | 78.47 | 78.64 | 0.86 |
| kNN | 82.06 | 80.54 | 83.96 | 80.20 | 82.22 | 0.89 |
| SVM | 59.58 | 55.72 | 88.39 | 31.46 | 68.35 | 0.78 |
| Random Forest | 97.02 | 97.57 | 96.36 | 97.66 | 96.96 | 0.99 |
| Gradient Boosting | 95.93 | 95.84 | 95.92 | 95.93 | 95.88 | 0.99 |
| CatBoost | 97.20 | 97.50 | 96.81 | 97.57 | 97.15 | 0.99 |
| **XGBoost** | 97.24 | 97.33 | 97.07 | 97.40 | 97.20 | 0.99 |
| Neural Network | 69.46 | 62.50 | 95.39 | 44.16 | 75.52 | 0.70 |

**Table 4**

**Mendeley Dataset classification performance after preprocessing**

| **Models** | **Accuracy (%)** | **Precision (%)** | **Sensitivity (%)** | **Specificity (%)** | **F-score (%)** | **AUC** |
|---|---|---|---|---|---|---|
| Logistic Regression | 94.79 | 93.91 | 95.65 | 93.94 | 94.77 | 0.98 |
| kNN | 94.66 | 96.32 | 92.73 | 96.54 | 94.49 | 0.97 |
| SVM | 96.63 | 96.96 | 96.19 | 97.06 | 96.57 | 0.99 |

*Contd...*

| | | | | | | |
|---|---|---|---|---|---|---|
| Random Forest | 96.85 | 97.06 | 96.54 | 97.14 | 96.80 | 0.99 |
| Gradient Boosting | 95.88 | 95.59 | 96.10 | 95.67 | 95.84 | 0.99 |
| **CatBoost** | 97.28 | 97.59 | 96.89 | 97.66 | 97.24 | 0.99 |
| XGBoost | 97.24 | 97.08 | 97.34 | 97.14 | 97.21 | 0.99 |
| Neural Network | 96.10 | 96.09 | 96.01 | 96.19 | 96.05 | 0.99 |

**Table 5**

**URL tokenized dataset with normalization applied classification performance**

| Models | Accuracy (%) | Precision (%) | Sensitivity (%) | Specificity (%) | F-score (%) | AUC |
|---|---|---|---|---|---|---|
| Logistic Regression | 89.10 | 88.59 | 89.45 | 88.76 | 89.02 | 0.95 |
| kNN | 87.44 | 89.94 | 83.96 | 90.83 | 86.85 | 0.93 |
| SVM | 91.46 | 90.46 | 92.47 | 90.49 | 91.45 | 0.97 |
| Random Forest | 91.29 | 91.22 | 91.14 | 91.44 | 91.18 | 0.97 |
| Gradient Boosting | 85.43 | 82.83 | 88.92 | 82.02 | 85.77 | 0.92 |
| CatBoost | 89.85 | 88.83 | 90.87 | 88.85 | 89.84 | 0.96 |
| XGBoost | 89.45 | 88.40 | 90.52 | 88.41 | 89.45 | 0.96 |
| **Neural Network** | 91.95 | 92.45 | 91.14 | 92.73 | 91.79 | 0.97 |

**Table 6**

**Merged Dataset (Mendeley dataset and tokenized URL) classification performance**

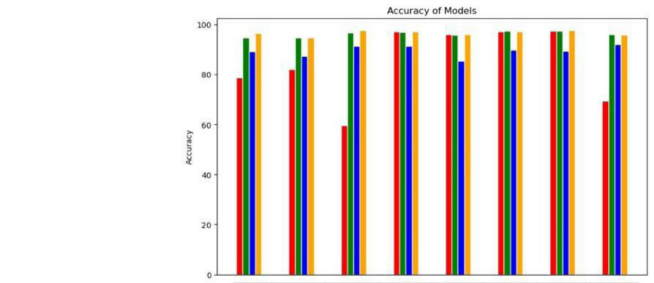
| Models | Accuracy (%) | Precision (%) | Sensitivity (%) | Specificity (%) | F-score (%) | AUC |
|---|---|---|---|---|---|---|
| Logistic Regression | 96.45 | 96.70 | 96.10 | 96.80 | 96.40 | 0.99 |
| kNN | 94.61 | 96.31 | 92.64 | 96.54 | 94.44 | 0.97 |
| **SVM** | 97.68 | 97.86 | 97.43 | 97.92 | 97.64 | 0.99 |
| Random Forest | 97.02 | 97.57 | 96.36 | 97.66 | 96.96 | 0.99 |
| Gradient Boosting | 96.10 | 96.34 | 95.74 | 96.45 | 96.04 | 0.99 |

*Contd...*

| | | | | | | |
|---|---|---|---|---|---|---|
| CatBoost | 97.11 | 97.41 | 96.72 | 97.49 | 96.06 | 0.99 |
| **XGBoost** | 97.68 | 97.95 | 97.34 | 98.01 | 97.64 | 0.99 |
| Neural Network | 95.88 | 95.75 | 95.92 | 95.85 | 95.84 | 0.99 |

On the original Mendeley dataset, XGBoost demonstrated the highest accuracy, achieving an impressive 97.24%. Additionally, both CatBoost and Random Forest yielded accuracy rates surpassing 97%. Notably, the Random Forest model outperformed other models in terms of precision and specificity on the same dataset. However, upon preprocessing the Mendeley dataset, CatBoost emerged as the leader with the highest accuracy at 97.28%. In this context, CatBoost not only secured the top spot in accuracy but also showcased superior precision, sensitivity, and F-score compared to all other employed machine learning models. Despite a marginal reduction in accuracy for some models post-preprocessing, the performance metrics on the original dataset exhibited significant variability, with certain models achieving below 80% accuracy. Conversely, on the preprocessed Mendeley dataset, all models consistently achieved accuracy levels higher than 94%, and none of the models fell below 90% in precision, sensitivity, specificity, or F-score. This underscores the substantial impact of preprocessing in enhancing model performance. The tokenized URLs, when compared to the Mendeley dataset, demonstrated comparatively lower performance. The highest accuracy attained was 91.95%, with Neural Network models exhibiting the best performance across all other metrics. This discrepancy highlights the inherent challenges associated with tokenized URLs. The combined dataset proved to be optimal for most models, as many models achieved their overall highest accuracy on it. Notably, XGBoost and SVM stood out by attaining the highest overall accuracy within the study, with 97.68%. Additionally, XGBoost, SVM, and Random Forest models consistently demonstrated accuracy rates surpassing 97%, showcasing their robust performance. All models showcased excellent performance, the average accuracy being 96.56%. SVM emerged as the leader in overall sensitivity and specificity, underlining its prowess in capturing both true positives and true negatives. Meanwhile, XGBoost excelled in achieving the highest overall precision, emphasizing its capability to minimize false positives. This collective performance underscores the effectiveness of the merged dataset in enhancing the models' overall predictive capabilities. Conclusively, it can be asserted that the amalgamation of datasets facilitated a significant

improvement in accuracy, sensitivity, specificity, and precision. The merged dataset played a pivotal role in refining the models' predictive capabilities and can be deemed valuable in advancing the identification of phishing websites. In Figure 4, the accuracy of all models is comprehensively depicted across various dataset variations. Figures 5, 6, 7, and 8 present the confusion matrices for each model on the Mendeley dataset, preprocessed Mendeley dataset, URL tokenized dataset, and merged dataset, respectively. These matrices offer a detailed insight into the models' performance, highlighting their ability to correctly classify instances and identify potential shortcomings. The ROC curve and Precision-Recall Curve for each model across different datasets are illustrated in Figures 9, 10, 11, and 12. An intriguing observation emerges from the confusion matrices, where the rate of False Negatives, denoting legitimate websites misclassified as phishing sites, tends to be higher than False Positives in some instances. This phenomenon prompts a deeper exploration into the model behavior and raises questions regarding the potential implications of this bias. Further investigation into the reasons behind this asymmetry can shed light on areas for model refinement and optimization. This imbalance in misclassifications suggests the need for a more nuanced understanding of the underlying patterns and features that contribute to this particular type of error, ultimately contributing to the continuous improvement of the models employed in the study.




| | LR | kNN | SVM | RF | GB | CB | XGB | NN |
|---|---|---|---|---|---|---|---|---|
| Mendeley Dataset | 78.78 | 82.06 | 59.58 | 97.02 | 95.93 | 97.20 | **97.24** | 69.46 |
| Preprocessed Mendeley Dataset | 94.79 | 94.66 | 96.63 | 96.85 | 95.88 | **97.28** | 97.24 | 96.10 |
| URL tokenized Dataset | 89.10 | 87.44 | 91.46 | 91.29 | 85.43 | 89.85 | 89.45 | **91.95** |
| Merged Dataset | 96.45 | 94.61 | **97.68** | 97.02 | 96.10 | 97.11 | **97.68** | 95.88 |

| Abbreviation | Full Form |
|---|---|
| LR | Logistic Regression |
| kNN | k-Nearest Neighbors |
| SVM | Support Vector Machine |
| RF | Random Forest |
| GB | Gradient Boosting |
| CB | CatBoost |
| XGB | XGBoost |
| NN | Neural Network |

**Figure 4**

**All models' (used in this study) accuracies across different dataset variations**

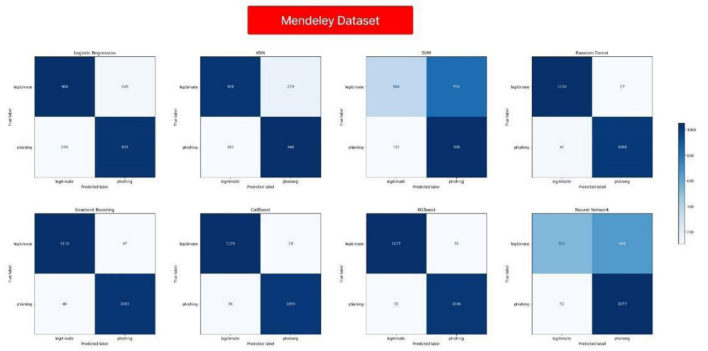



**Figure 5**

**Mendeley dataset confusion matrix**

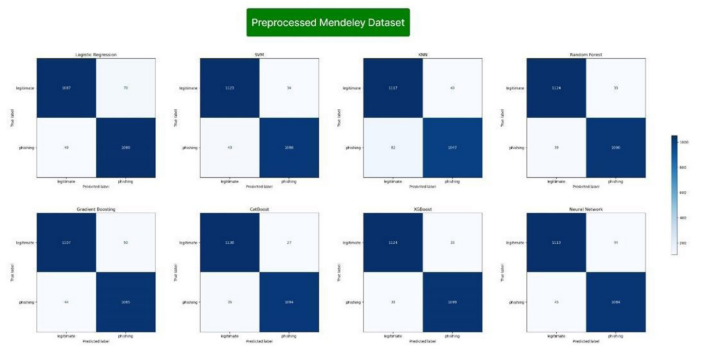



**Figure 6**

**Preprocessed Mendeley dataset confusion matrix**

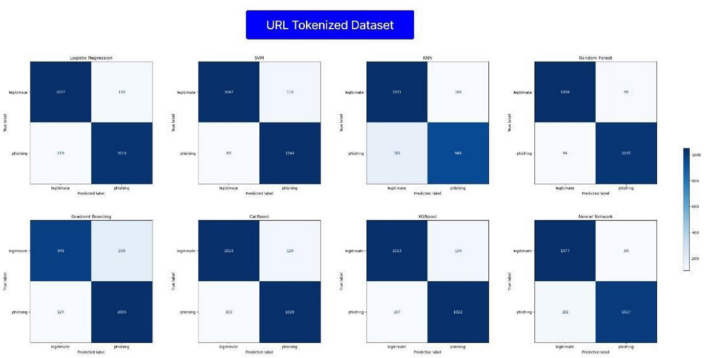



**Figure 7**

**URL tokenization confusion matrix**

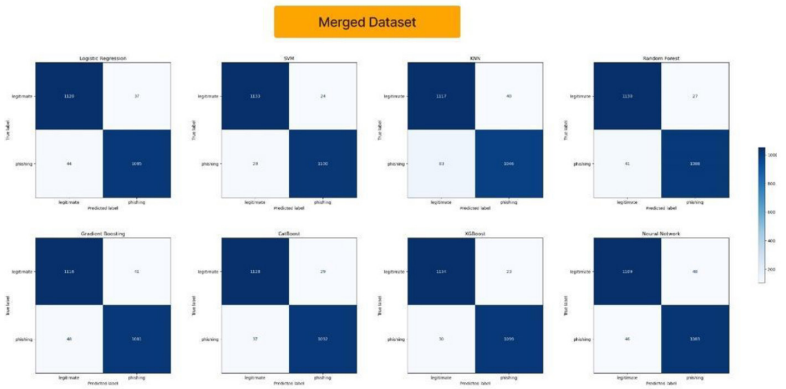



**Figure 8**

**Merged dataset confusion matrix.**

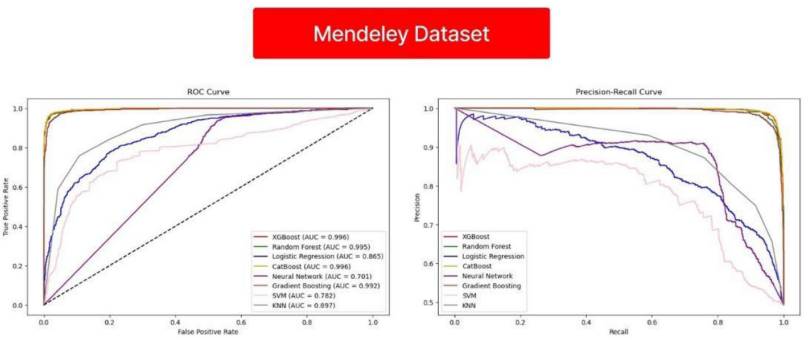



**Figure 9**

**Mendeley dataset ROC curve and Precision-Recall curve**

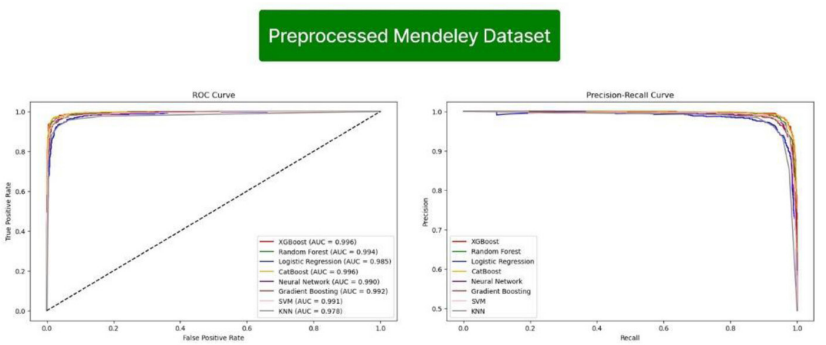



**Figure 10**

**Preprocessed Mendeley dataset ROC curve and Precision-Recall curve**

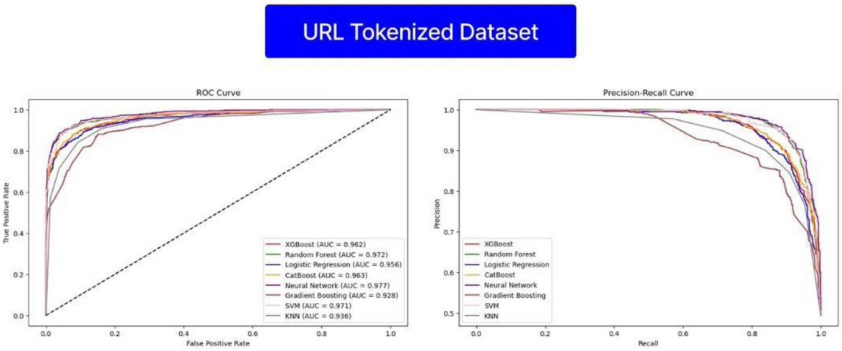



**Figure 11**

**URL tokenization ROC curve and Precision-Recall curve**

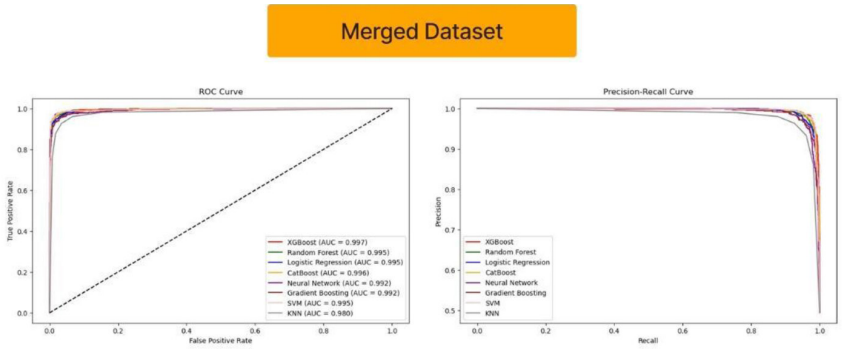



**Figure 12**

**Merged dataset ROC curve and Precision-Recall curve**

The results section unveils the performance metrics of machine learning models on distinct datasets. Notably, CatBoost exhibits superior accuracy and precision, outperforming other models on the preprocessed Mendeley dataset. The tokenized URLs, while demonstrating lower performance, provide valuable insights into the challenges associated with this approach. The combined dataset proves optimal for many models, with XGBoost and SVM leading in overall accuracy. Confusion matrices, ROC curves, and Precision-Recall curves offer nuanced perspectives on model behavior, emphasizing the need for continuous refinement and investigation into misclassification patterns.

## 5. Conclusion

In conclusion, this study emphasizes the ongoing threat of phishing attacks in the cybersecurity landscape, necessitating continuous advancements in detection methods. The research employs three datasets, namely the Mendeley dataset, URL tokenized dataset, and a merged dataset combining Mendeley and URL tokenized data. Multiple machine learning algorithms, including LR, kNN, SVM, RF, GBM, NN, CatBoost, and XGBoost, are utilized. On the original Mendeley dataset, XGBoost achieves the highest accuracy at 97.24%, with CatBoost and Random Forest also surpassing 97%. Post-preprocessing, CatBoost leads with an accuracy of 97.28%, showcasing superior precision, sensitivity, and F-score compared to other models. Despite slight accuracy reductions post-preprocessing, models consistently achieve over 94% accuracy on the preprocessed Mendeley dataset, highlighting the substantial impact of preprocessing on performance. Tokenized URLs exhibit comparatively lower performance, with the highest accuracy at 91.95%, emphasizing the challenges associated with this approach. The combined dataset proves optimal for most models, with XGBoost and SVM achieving the highest overall accuracy at 97.68%. SVM excels in sensitivity and specificity, while XGBoost excels in precision. The merged dataset significantly enhances accuracy, sensitivity, specificity, and precision, underscoring its pivotal role in refining predictive capabilities for identifying phishing websites. The results section provides a detailed overview of machine learning model performance on different datasets. CatBoost emerges as a standout performer on the preprocessed Mendeley dataset. The tokenized URLs offer valuable insights into associated challenges, and the combined dataset proves effective for various models. Confusion matrices, ROC curves, and Precision-Recall curves provide nuanced perspectives on model behavior, emphasizing the need for ongoing refinement and investigation into misclassification patterns to enhance model effectiveness in combating phishing threats.

## References

[1] Gandotra, E., & Gupta, D., An efficient approach for phishing detection using machine learning. *Multimedia Security: Algorithm Development, Analysis and Applications*, 239-253 (2021).

[2] Omar, Asmaa Reda, Shereen Taie, and Masoud E. Shaheen. "From Phishing Behavior Analysis and Feature Selection to Enhance Pre-

diction Rate in Phishing Detection." *International Journal of Advanced Computer Science and Applications* 14.5 (2023).

[3] Uddin, Md Milon, et al. "A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information." 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI). IEEE (2022).

[4] Ojewumi, T. O., Ogunleye, G. O., Oguntunde, B. O., Folorunsho, O., Fashoto, S. G., & Ogbu, N. J. S. A., Performance evaluation of machine learning tools for detection of phishing attacks on web pages. *Scientific African*, *16*, e01165 (2022).

[5] Jain, A. K., & Gupta, B. B., Comparative analysis of features based machine learning approaches for phishing detection. In 2016 3rd international conference on computing for sustainable global development (INDIACom)(pp. 2125-2130). IEEE (2016, March).

[6] Niakanlahiji, A., Chu, B. T., & Al-Shaer, E., Phishmon: A machine learning framework for detecting phishing webpages. In 2018 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 220-225). IEEE (2018, November).

[7] Deshpande, A., Pedamkar, O., Chaudhary, N., & Borde, S., Detection of phishing websites using Machine Learning. *International Journal of Engineering Research & Technology (IJERT)*, *10*(05) (2021).

[8] Almseidin, M., Zuraiq, A. A., Al-Kasassbeh, M., & Alnidami, N., Phishing detection based on machine learning and feature selection methods (2019).

[9] Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W., A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27-39 (2019).

[10] Hannousse, A., & Yahiouche, S., Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence*, *104*, 104347 (2021).

[11] Mohamed, G., Visumathi, J., Mahdal, M., Anand, J., & Elangovan, M., An effective and secure mechanism for phishing attacks using a machine learning approach. *Processes*, *10*(7), 1356 (2022).

[12] Abdelhamid, N., Thabtah, F., & Abdel-Jaber, H., Phishing detection: A recent intelligent machine learning comparison based on models content and features. In 2017 IEEE international conference on intelligence and security informatics (ISI) (pp. 72-77). IEEE (2017, July).

[13] Amit Kumar Gupta, Pushpa Gothwal, Dinesh Goyal & Carlos M. Travieso-Gonzalez. IoT-Galvanized pandemic special E-Toilet for genera-

tion of sanitized environment, *Journal of Discrete Mathematical Sciences and Cryptography* (2022), DOI: 10.1080/09720529.2022.2068607.

[14] Amit Kumar Gupta, Vijander Singh, Priya Mathur & Carlos M. Travieso-Gonzalez. Prediction of COVID-19 pandemic measuring criteria using support vector machine, prophet and linear regression models in Indian scenario, *Journal of Interdisciplinary Mathematics* (2020), DOI: https://doi.org/10.1080/09720502.2020.1833458.

[15] Kumar, Anil , Gupta, Amit Kumar, Panwar, Deepak , Chaurasia, Sandeep & Goyal, Dinesh. Operating system security with discrete mathematical structure for secure round robin scheduling method with intelligent time quantum, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1519–1533 (2023), DOI: https://doi.org/10.47974/JDMSC-1816.

[16] Joshi, Ruchi, Mathur, Priya, Gupta, Amit Kumar, Singh, Suyesha, Paliwal, Vismita & Nayar, Sejal. Mathematical modeling of intelligent system for predicting effectiveness of premenstrual syndrome, *Journal of Interdisciplinary Mathematics*, 26:3, 551-562 (2023), DOI: https://doi.org/10.47974/JIM-1681 (Q2 Journal).

[17] Hannousse, Abdelhakim; Yahiouche, Salima, "Web page phishing detection", Mendeley Data, V3 (2021), doi: 10.17632/c2gw7fy2j4.3.

[18] Anil Kumar & Sandeep Kumar Sharma. Information cryptography using cellular automata and digital image processing, *Journal of Discrete Mathematical Sciences and Cryptography*, 25:4, 1105-1111 (2022), DOI: 10.1080/09720529.2022.2072437.

[19] Vijayarani, S., and R. Janani. "Text mining: open source tokenization tools-an analysis." *Advanced Computational Intelligence: An International Journal (ACII)* 3.1 : 37-47 (2016).

[20] Sharma, Sandeep Kumar, Kumar, Anil, Ashtagi, Rashmi & Jain, Rekha. OCA: An intelligent model for improving security breach of biometric based authentication systems, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1415–1425 (2023), DOI: 10.47974/JDMSC-1765.

[21] Liu, Huan, and Rudy Setiono. "Chi2: Feature selection and discretization of numeric attributes." Proceedings of 7th IEEE international conference on tools with artificial intelligence. Ieee (1995).

[22] Mahesh, Batta. "Machine learning algorithms-a review." *International Journal of Science and Research (IJSR).[Internet]* 9.1 : 381-386 (2020).

[23] Dasgupta, Dipankar, Zahid Akhtar, and Sajib Sen. "Machine learning in cybersecurity: a comprehensive survey." *The Journal of Defense Modeling and Simulation* 19.1 : 57-106 (2022).

[24] Hannousse, Abdelhakim, and Salima Yahiouche. "Towards benchmark datasets for machine learning based website phishing detection: An experimental study." *Engineering Applications of Artificial Intelligence* 104 : 104347 (2021).

[25] Nick, Todd G., and Kathleen M. Campbell. "Logistic regression." *Topics in biostatistics* : 273-301 (2007).

[26] Guo, Gongde, et al. "KNN model-based approach in classification." On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7 (2003). Proceedings. Springer Berlin Heidelberg, 2003.

[27] Suthaharan, Shan, and Shan Suthaharan. "Support vector machine." Machine learning models and algorithms for big data classification: thinking with examples for effective learning : 207-235 (2016).

[28] Kecman, Vojislav. "Support vector machines–an introduction." Support vector machines: theory and applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 1-47 (2005).

[29] Yu, Hwanjo, and Sungchul Kim. "SVM Tutorial-Classification, Regression and Ranking." Handbook of Natural computing 1 : 479-506 (2012).

[30] Qi, Yanjun. "Random forest for bioinformatics." Ensemble machine learning: Methods and applications : 307-323 (2012).

[31] Natekin, Alexey, and Alois Knoll. "Gradient boosting machines, a tutorial." *Frontiers in Neurorobotics* 7 : 21 (2013).

[32] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems 31 (2018).

[33] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (2016).

[34] Popescu, Marius-Constantin, et al. "Multilayer perceptron and neural networks." *WSEAS Transactions on Circuits and Systems* 8.7 : 579-588 (2009).

[35] Li, Hao, et al. "Rethinking the hyperparameters for fine-tuning." arXiv preprint arXiv:2002.11770 (2020).