CLASSIFICATION OF PHISHING ATTACKS USING MACHINE LEARNING ALGORITHMS: A SYSTEMATIC LITERATURE REVIEW

Abstract

Phishing attacks have been a major threat tocyber security since they take advantage of human vulnerabilities rather than system setbacks, making them difficult to detect. Phishing attacks always involve fraudulent websites designed to mimic legitimate websites to steal sensitive information from victims. This research paper provides a comprehensive literature review to recommend future research.This review paper examinesprevious papers' application of machine learning (ML) algorithms to phishing detection, focusing on how ML can be used to turn phishing attack problems into classification tasks. This research compared the commonly used ML algorithms like Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), Naïve Bayes (NB), k-means Clustering, and Artificial Neural Networks (ANN), these algorithms were compared based on their performance, strengths, and weakness.Key findings reveal that SVM excels with high-dimensional data, RF handles large datasets efficiently, and DT offers simplicity but struggles with complex features. Algorithm performance depends on data and feature selection.

. This presents the need to develop hybrid or ensemble models to improve detection accuracy and reliability and contribute to stronger cybersecurity frameworks.

Keywords: Machine Learning; phishing attacks; literature review; algorithms, models.

1. Introduction

Securing the Network in an organization is a crucial issue that should be taken into consideration. One way to secure a network is to authorize individual accounts, which involves using usernames and passwords to prevent illegal access to a particular account (Faris et al., 2023). This authentication process is required to prevent unauthorized access to sensitive data. Despite a good and secure network, there are still situations where unauthorized people access sensitive data, and a common method used by these hackers is phishing attacks.Phishing attacks are the process where attackers create fake websites that imitate a legitimate website, this is done to get sensitive information from victims, and use the information for criminal purposes like illegal financial gain (Almousa et al., 2022). These attackers always send a Uniform Resource Locator (URL) link that looks authentic to the victims, asking them to update or confirm their information by clicking on it (Shantanu & Joshua, 2021). Phishing emails are often used to lure individuals to the compromised websites to request their personal information, such as details of their bank account, which the attacker will use to steal sensitive data that the victim of the attack has submitted (Guaña-Moya et al., 2022). Phishing attacks are always associated with spam emails, which may contain links that will redirect victims to phishing websites. Phishing attacks are difficult to detect, as the location of the server is always disguised and the URL of the phishing website always looks like a legitimate website, it is difficult for good security software to detect these websites because they don’t rely on the computer's malware infection (Azzani et al., 2024).

Researchers have proposed a lot of work on detecting phishing attacks in the literature and commercial products. Figure 1 shows the four main features that can be used in the detection of phishing attacks. One of the features is the URL-basedfeature, this feature works based on the URL. The URL which is a phishing link directs a victim to a specific page that is a duplicate of the original. The URL length, the count digit in the URL, and the correct spelling of the URL can all be used to distinguisha malicious URL from a legitimate URL. Another feature that can be used in the detection of phishing attacks is the domain-based feature. This feature works by identifying if a URL is a phishing URL or not based on the domain name. The third feature that can be used in the detection of phishing attacks is the page-based feature, which works by using the information from the pages to determine the reputation ranking services. The fourth feature is the content-based feature, which works based on the scanning process of the domain, the content-based feature scans the page title, hidden text, meta tags, body texts, and images in the page to determine whether the page requires the login process, the category of the page, and the user.

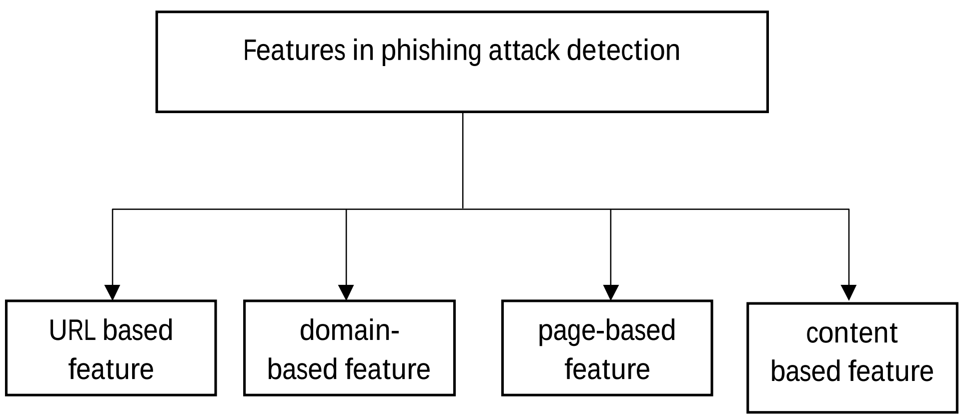

Figure 1. Features in phishing attack detection (Jupin et al., 2019)

The four highlighted features are commonly used to identify phishing attempts. However, the feature might not alwaysidentify phishing attempts effectively due to their individual limitations.Thus, selecting a featureshould be based on the detection mechanismand carefully picked.

Phishing detection isdifficult because of the way attackers explore human vulnerabilities and not system errors. Phishing detection is classified as a classification problem, meaning that a suspected page needs to be labeled as legitimate or phishing. Thus, a good and reliable method is needed for detection. The machine learning method has proven to be a good and reliable method for the detection of phishing attacks over the years, due to its ability to transform phishing attack problems into classification tasks. Machin learning is a subset of artificial intelligence (AI), and its goal is to allow computers to learn from historical data to make decisions based on patterns. Machine learning works by training an algorithm with the uses of the dataset with specific features. In the case of phishingdetection, features from URL, domain, page, or content are being used to detect if a web page is legitimate or fake. This method is good for phishing attack detection since it converts the detection problem into a classification task. Many machine learning algorithms have been used in the detection of phishing attacks over the years, but the widely used algorithms currently are the Artificial Neural Network (ANN) algorithm, k-means clustering, Support Vector Machine (SVM), Naïve Bayes (NB), Decision Tree (DT), and Random Forest (RF) algorithm. These methods were chosen because of their performance and high accuracy in detecting phishing attacks.This research will evaluate and compare these algorithms to enhance phishing detection, by focusing on accuracy, efficiency, and feature optimization for improved cybersecurity.

2. Literature Review

Thakur et al. (2023) carried out a systematic literature review on the deep learning methods used in phishing detection, the research reviewed relevant papers to identify deep learning techniques, their effectiveness, strengths, and limitations. The study shows that deep learning-based algorithms like NNs and LSTMs are mostly used for phishing detection. The research highlighted strengths, limitations, and gaps, and proposed future research directions to improve detection accuracy.

Do et al. (2022) conducted a systematic literature review on deep learning techniques for phishing detection. The research uses the systematic literature review approach to classify and analyze deep learning techniques using a proposed taxonomy; and empirical evaluation to assess performance and identify. the result shows that the research identified taxonomies of phishing detection and deep learning algorithms, analyzed advantages and disadvantages, and highlighted issues like manual parameter-tuning, long training times, and deficient detection accuracy.

Adane&Beyene (2022)reviewed recent studies on the application of machine learning and deep learning-based techniques fordetecting phishing attacks on websites. The result of the systematic literature review Identified research gaps, including imbalanced datasets, improper dataset sourcing, unjustified dataset splits, disputes on feature selection, lack of consensus on phishing website lifespans, small dataset definitions, and run-time issues. This provided a structured summary for future research guidelines.

Wood et al. (2022) carried out a systematic literature review of anti-phishing defenses, with a focus on before-the-click detection techniques and their application to phishing emails. The research reviewd21 primary studies and 335 secondary studies. The result from the study shows the existing techniques, and their performance when used for phishing email detection, the study also suggested promising areas for future research.

Saraswathi et al. (2023) Reviewed phishing detection techniques and proposed the use of algorithms like ANN, SVM, RF, and K-NN, trained and tested on publicly available datasets (e.g., UCI ML repository). The research further developed a framework for accurate phishing website detection and highlighted the use of ML algorithms to classify URLs effectively.

3. Methodology

The method that was adopted in this research involved a comprehensive review and comparative analysis of machine learning methods for phishing attack detection. The method follows a specific step which includes:

3.1 Literature Survey

We searched four different academic databases to collect journal papers and conference proceedings on the application of ML algorithms such as DT, RF, SVM, NB, k-means clustering, and ANN on phishing detection, focusing on key features such as URL, domain, and page.The academic databases that were considered in this research are Science Direct, Google Scholar, Scopus, and Web of Science.

The search string that was used to search for the research papers from the academic databases were “phishing detection”, “URL classification”, Machine learning methods for phishing detection”, “decision tree approach for phishing detection”, “random forest approach for phishing detection”, “Support vector machine approach for phishing detection”, “Naïve bayes approach for phishing detection”, k-means clustering approach for phishing detection”, “artificial neural network approach for phishing detection”, and “strategies for selecting algorithm in machine learning”.

3.2 Data retrieval

After the first search, we retrieved 415 research papers from the databases. This was done by searching with the search words and also joining the search words using Boolean 'OR'. Table 1 shows the paper collection and screening process, from each database. We further streamlined the search to only computer science-related papers and papers that were published from 2018 to 2024. At this point, 184 papers remain, as 231 papers were removed. After the first screening, we went through the remaining 184 paper’s abstracts to screen out those papers that were not relevant to the research and those that did not meet the inclusion criteria of the research. After this second screening, we discovered that only 50 papers met the inclusion criteria and were relevant to the research, these papers further underwent a quality evaluation to achieve the aim of this research. Table 1: first search result

	Science Direct	Google Scholar	, Scopus	Web of Science	Total
First search	145	168	46	56	415
First screening	53	58	27	46	184
Second screening	12	21	7	10	50

3.3 Eligibility criteria

Inclusion criteria: The inclusion criteria for selecting the research papers include journal papers and conference proceedings, that were published from 2018 to 2024. Papers written in the English language and papers related to the application of machine learning and deep learning on phishing detection were all included in the research. Additionally, in situations where we have papers with identical studies and outcomes, we chose the most recent paper.

Exclusion criteria: Papers that were excluded in this research include papers that are written in other languages apart from the English language. Also, we excluded papers that are not related to phishing detection and papers whose contributions to the work are not explicitly stated in the abstract.

3.4 Feature Examination

WeIdentified how each ML algorithm leverages features for phishing detection, such as URL length, domain reputation, page structure, and embedded content.

3.5 Algorithm Evaluation

We collected phishing detection experimental results from existing studies, and compared their performances based on their accuracy, efficiency, computational complexity

3.6 Comparative Analysis

We created a table to compare the studied algorithms in terms of the strengths and weaknesses of the algorithms.

3.7 Recommendations

We proposed areas for future research, including the development of ensemble models and adaptive frameworks for real-time phishing detection.

By adopting this methodology, weensure a detailed analysis of the selected ML models, providing insights into their effectiveness in the detection of phishing attacks.

4. Result

4.1 Machine learning-based method in classifying phishing attacks

a. Decision Tree (DT) Algorithm

The DT algorithm is a supervised classification machine learning algorithm. This machine-learning algorithm can be used to solve both regression and classification problems. The DT algorithms have two types which are the Iterative Dichotomiser 3 (ID3) and the C4.5 algorithms. The uses a “top-down” method to create an addiction tree, and has proven effective over the years. However, it has a lot of setbacks which can affect its application in real-life situations(Charbuty&Abdulazeez, 2021). The C4.5 on the other hand was developed to address the setbacks of the ID3 algorithm, and it has proven to be a better solution when using large and noisy data. The DT algorithm can be expressed mathematically by describing its key concepts which are entropy, information gain, and the recursive partitioning process to split the data:

Entropy

For a dataset S with classes $c_1, c_2, c_3, \ldots, c_n$, the entropy $E(S)$ is given by:

$$E(S) = -\sum_{i=1}^{n} p\,(c_i)\log_2\big(p(c_i)\big) \qquad \ldots 1$$

were

$p(c_i)$= probability of class c_i in S.

Information gain (IG)

The $IG(S, A)$ when splitting an attribute A is given by:

$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} E(S_v) \qquad \ldots 2$$

where:

S_v= subset of S for which attribute A has value v,
$\frac{|S_v|}{|S|}$= proportion of examples in S with value v for attribute A,
$E(S_v)$= entropy of subset S_v.

Recursive Splitting

To build the tree, it selects the attribute A_j with the highest IG:

$$A_{\text{best}} = \underset{A}{\operatorname{argmax}} IG(S, A) \qquad \ldots 3$$

Then split S based on A_{best}, and recursively repeat for each subset S_v until a stopping criterion is met.

Stopping Criterion

Stop splitting when:

$$E(S) = 0 \quad \text{or when other criteria are met (e.g., max depth)} \ldots 4$$

Prediction Rule

To classify a new instance x in a trained DT, follow decision rules at each node based on attribute values $A_i(x)$ until reaching a leaf node that provides the prediction (Mitra & Padmanabhan, 2023).

Ganesan, (2022) researchedwebsite-based phishing detection using the C4.5 algorithm, the dataset that was used in the research has about 300 websites. The result from their work the proposed algorithm was able to outperform other compared algorithms by achieving the highest accuracy of 90.8% when evaluated with the use of the confusion matrix. On the other hand, Sankhyan et al. (2023) proposed a phishing detection method with the use of the ID3 algorithm. The research method used for the implementation has four main steps: data preparation, feature extraction, implementation, and evaluation.

b. K-Means Clustering

The k-means clustering algorithm is an unsupervised machine learning algorithm that works by partitioning data points into different clusters of similar data points. The k-meant algorithm is used to partition n data points into k clusters, where each observation belongs to a cluster of the nearest mean (Sinaga& Yang, 2020). This algorithm's objective is to minimize the variance within each cluster by updating centroids iteratively. The algorithm can be represented mathematically as:

I. **Initialize** k centroids, one for each cluster.

II. **Assignment Step**: For each data point xi in the dataset, assign it to the nearest centroid µj . This assignment is based on minimizing the distance between xi and µj, typically using the Euclidean distance:

$$c_i = \underset{j}{\operatorname{argmin}} \parallel x_i - \mu_j \parallel^2 \quad \ldots 5$$

where:

ci is the index of the centroid closest to xi.
$\parallel x_i - \mu_j \parallel^2$ denotes the squared Euclidean distance between the point xi and centroid µj.

III. **Update Step**: After assigning each point to a cluster, update each centroid µj as the mean of all points xi assigned to it:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \qquad \ldots 6$$

where:

Cj is the set of points assigned to cluster j.
|Cj| is the number of points in cluster j.

IV. **Objective Function**: The algorithm minimizes the sum of squared distances within each cluster.

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \parallel x_i - \mu_j \parallel^2 \qquad \ldots 7$$

The K-means algorithm repeats the assignment and update steps until J converges. The K-means algorithm finds clusters by minimizing J iteratively, aiming to group data points so that they are as close as possible to their assigned centroid (Harris & de Amorim, 2022).

Al-Sabbagh et al. (2024)researched phishing detection using the kernel k-means clustering algorithm, which is an extension of the k-means algorithm. The researchutilizes public dataset datasets of varying sizes (2000, 7000, and 10,000 samples). The result from the experiment shows that the proposed method outperformed the compared method with the highest accuracy of 89.2% on the 2000-sample dataset. Another research that was carried out by Arab &Sohrabi (2017) applied the k-means clustering algorithm to phishing detection. However, the researchers proposed four different algorithms: k-means clustering, J48 decision tree, multilayer perceptron (MLP),and Naïve Bayes.From the experimental result, the k-means clustering outperformed all other compared algorithms, achieving a prediction accuracy of 99%. However, the MLP algorithm has a lower production time as compared to the k-means clustering algorithm. Saputraet al. (2018) proposed another work on phishing detection with the use of the k-means algorithm. The classification was processed 10-fold and the result shows 96.49% accuracy and a 3.51% error rate.

c. Naïve Bayes (NB)

The NB algorithm, which is also referred to as the Bayesian classifier, is a probabilistic classifier based on Bayes' theorem with the "naïve" assumption of conditional independence between the features. This ML algorithm is mostly used for sentiment analysis, text classification, and spam filtering because of its simplicity and efficiency (Nakhipova, et al., 2024). NB can be represented mathematically as:

I. Bayes' Theorem: To determine the probability of a class C given a feature vector $X = (x_1, x_2, \dots, x_n)$. With the use of Bayes' theorem, this probability(posterior probability) is given by:

$$P(C \mid X) = \frac{P(X|C) \cdot P(C)}{P(X)} \qquad \dots 8$$

where:

$P(C \mid X)$= posterior probability of class C given features X.
$P(X \mid C)$= likelihood of observing X given class C.
$P(C)$= prior probability of class C.
$P(X)$= evidence or the total probability of observing X across all classes.

II. Naïve Bayes Classifier Assumptions

Conditional Independence: The NB algorithm assumes that each feature x_i is conditionally independent of every other feature x_j00 given the class C. This simplifies the likelihood calculation:

$$P(X \mid C) = P(x_1 \mid C) \cdot P(x_2 \mid C) \cdots P(x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C) \quad \dots 9$$

Class Prediction: To classify a new instance, the NB algorithm assigns it to the class C that maximizes the posterior probability $P(C \mid X)$. Since $P(X)$ is constant for all classes, it can be ignored in the maximization:

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(C \mid X) = \underset{C}{\operatorname{argmax}}\left(P(C) \prod_{i=1}^{n} P\left(x_i \mid C\right)\right) \quad \text{...10}$$

Summary of Naïve Bayes Mathematical Steps

Compute the Prior Probability $P(C)$ for each class C:

$$P(C) = \frac{\text{Number of instances in class } C}{\text{Total number of instances}} \quad \text{...11}$$

Compute the Likelihood $P(x_i \mid C)$ for each feature x_i given class C:

This is typically estimated from the training data and varies based on the type of Naïve Bayes used (Gaussian, Multinomial, or Bernoulli).

Compute the Posterior Probability $P(C \mid X)$ using Bayes' theorem:

$$P(C \mid X) \propto P(C) \prod_{i=1}^{n} P\left(x_i \mid C\right) \quad \text{...12}$$

Prediction: Choose the class $\hat{C}$ that maximizes the posterior probability:

$$\hat{C} = \underset{C}{\operatorname{argmax}}\left(P(C) \prod_{i=1}^{n} P\left(x_i \mid C\right)\right) \quad \text{...13}$$

Example of Naïve Bayes Variants

V. **Gaussian Naïve Bayes**: For continuous features, assuming a Gaussian distribution:

$$P(x_i \mid C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i-\mu_C)^2}{2\sigma_C^2}\right) \quad \text{...14}$$

where μ_C and σ_C are the mean and standard deviation of the feature x_i for class C.

VI. **Multinomial Naïve Bayes**: For discrete features (e.g., word counts in text classification):

$$P(x_i \mid C) = \frac{\text{Count of } x_i \text{ in class } C+1}{\text{Total count of all features in class } C+V} \quad \text{...15}$$

where$V$ is the vocabulary size

VII. **Bernoulli Naïve Bayes**: For binary features:

$$P(x_i \mid C) = \begin{cases} p_{i,C} & \text{if } x_i = 1, \\ 1 - p_{i,C} & \text{if } x_i = 0 \end{cases} \qquad \ldots 16$$

where$p_{i,C}$ is the probability of a feature x_i appearing in class C (Pajila et al., 2023).

Krishna, (2021) carried out research on phishing detection in spam emails with the use of the NB classifier as the text classification method. The method used divided words into tokens that represent works used in non-spam and spam emails. The NB was used to classify phishing web pages in the work, features such as URL, source, and images. The researchers used spam filtering techniques to protect mailboxes for spam mail the result shows an accuracy of more than 80%. Anotherwork that was done by Singh (2019), proposed the use of NB to classify emails an legit or fake, the researchers used the intelligent water drop algorithm to perform the feature selection task, and the result from the experiment shows the ability of the NB In phishing detection, as the proposedmodel was able to achieve a high accuracy more than 80%.

d. Random Forest (RF) Algorithm

The RF algorithm is an ensemble of different decision trees for classification and regression purposes. The RF algorithm works by building multiple trees with the use of bootstrapped samples and aggregating the results (Team, 2023). The algorithm can be represented mathematically as shown below:

Given a dataset (D) with samples (n) and features (m), Random Forest builds T decision trees, each trained on a random subset of the data. It uses Bagging and Random Feature Selection to build each tree, which helps reduce overfitting and improve model accuracy.

The process involves the following steps:

I. Bootstrapping the Dataset

Each tree t is trained on a bootstrap sample D_t of the original dataset D. A bootstrap sample is created by randomly sampling n examples from D with replacement

II. Building Each DT with Random Feature Selection

For each decision tree:

- At each node of the tree, rather than considering all m features, a random subset of k features is chosen, where k<m.
- The best feature among this subset is selected to split the node, based on some impurity measure.

For each decision tree in the forest, the following optimization is performed at each split node to minimize impurity:

For classification:

- Let G be the Gini impurity of a node:

$$G = \sum_{i=1}^{K} p_i (1 - p_i) = 1 - \sum_{i=1}^{K} p_i^2 \qquad \ldots 17$$

wherep_i is the proportion of samples in the class i at the node, and K is the number of classes.

For regression:

- Let MSE be the Mean Squared Error of a node:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2 \qquad \ldots 18$$

- where y_i is the actual value of the target variable for the i-th sample, $\bar{y}$ is the mean target value for all samples at that node, and n is the number of samples in the node.

The algorithm splits the node using the feature that minimizes the impurity after the split

Once all T trees are built, the Random Forest makes predictions by aggregating the outputs of these trees:

- For classification:

Each tree casts a “vote” for a class C_j.
The final prediction is determined by majority voting:

$$\hat{y} = \underset{j}{\operatorname{argmax}} \sum_{t=1}^{T} 1 \left\{y_t = C_j\right\} \qquad \ldots 19$$

where 1 is the indicator function that equals 1 if $y_t = C_j$ (i.e., tree t predicted class C_j) and 0 otherwise.

- For regression:

The final prediction is the average of the predictions from all trees:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} y_t \qquad \ldots 20$$

wherey_t is the predicted value from tree t (Wu et al., 2023).

Somesha&Pais (2022)carried out research that applied the RF algorithm to phishing detection. The researchers aim to compare six different machine learning algorithmsin the detection of phishing attacks. The research made use of a real-time input dataset to enhance accuracy in email anti-phishing solutions.The result from the experiment shows that the RF outperformed all other compared algorithms with the highest accuracy of 99.50%. Rajoju et al. (2024)carried out another research that applied the RF algorithm to the detection of phishing attacks. The research also applied other machine learning algorithms which include the Naive Bayes, Decision Trees, Logistic Regression, Random Forest, AdaBoost, and KNN. The result of the experiment shows that the RRF outperformed all other compared algorithms, by achieving the highest accuracy of accuracy rate of 96%.Jagadeesan et al. (2018) also researched the detection of phishing attacks in URLs using RF. The research used the metadata of the URL such as the number of slashes and keywords in the URL portion. They further used the Rf algorithm for the classification of URLs as legitimate or phishing attacks. Two different datasets were used for the research including a dataset that had 2500 instances with 31 different attributes and another with1353 instances with 31 different attributes

e. Support Vector Machine (SVM) Algorithm

The SVM algorithmis a supervisedML algorithm used mostly for classification problems. The algorithm works by classifying datasets containing class labels and features (Saini, 2024). The mathematical representation of the SVM algorithm is shownbelow:

I. Problem Setup and Hyperplane

In an SVM, we assume we have a set of training data:

$$(x_i, y_i) \text{for } i = 1, 2, \dots, N \qquad \dots 21$$

where:

- $x_i \in R^n$ = feature vector of the ii-th sample,
- $y_i \in \{-1, +1\}$ = class label, either -1 or +1.

The SVM's goal is to find a hyperplane that maximally separates the two classes.

The hyperplane in n-dimensional space can be defined as:

$$w \cdot x + b = 0 \qquad \dots 22$$

where:

- w = weight vector
- b = bias term.

II. Decision Boundary and Margin

The decision function is:

$$f(x) = w \cdot x + b \qquad \dots 23$$

For classification, the sign of $f(x)$ determines the class of x:

- If $f(x) > 0$, then x is classified as +1.
- If $f(x) < 0$, then x is classified as -1.

III. Optimization Objective

To maximize margin, an SVM optimization problem can be formulated. For a correctly classified point, the constraint is:

$$y_i(w \cdot x_i + b) \geq 1 \text{for } i = 1, 2, \dots, N \quad \dots 24$$

The margin width is $\frac{2}{\|w\|}$, so maximizing the margin is equivalent to minimizing $\| w \|$. The optimization problem becomes:

Primal Formulation (Hard Margin SVM)

$$\min_{w,b} \frac{1}{2} \| w \|^2 \qquad \ldots 25$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1, \forall i \qquad \ldots 26$$

Soft Margin SVM (for Non-Separable Data)

The slack variables $\xi_i \geq 0$ can be introduced for each data point, to handle situations where data isn't perfectly separable:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \qquad \ldots 27$$

The objective function is modified to penalize misclassifications:

$$\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i \qquad \ldots 28$$

where$C$ is a regularization parameter that balances maximizing the margin and minimizing the classification error.

IV. Dual Formulation (Lagrangian)

The above primal problem can be reformulated into its dual form, which is useful for using kernel functions in SVM. The dual form is:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \qquad \ldots 29$$

subject to:

$$0 \leq \alpha_i \leq C, \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \ldots 30$$

where α_i = Lagrange multipliers.

V. Decision Function

Once w and b are determined, the decision function for classifying a new sample x is:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b\right) \qquad \ldots 31 \text{ (Pirouz\&Pirouz, 2023)}$$

Zouina&Outtaj (2018) carried out research on phishing detection in URLs with the use of the SVM algorithm. The dataset used by the researcher contains 2000 instances with 1000 phishing URLs and 1000 legitimate URLs. The result from the experiment shows that the SVM was able to achieve 95.80% accuracy. another research that applied the SVM to phishing detection was carried out by Elsheh&Swayeb, (2023). These researchers used a hybrid model of SVM Ant Colony Optimization (ACO) to detect phishing in web content. The dataset used by the researcher contains 12,000 instances. The result from their experiment shows that the proposed model achieved 97.54% accuracy.

f. Artificial Neural Network (ANN)Algorithm

ANN algorithm is a deep learning algorithm that is structured from the human brain to stimulate human behavior. The ANN uses the neurons as its basic processing unit (Ferreira et al., 2018; Kalhor, 2020). Its neurons are linked together in different layers to make up the ANN. An ANN can be expressed mathematically as layers of neurons where each neuron computes

$$z = w^{\top}x + b \qquad \ldots 32$$

where

w = weight vector

x = input, and

b = bias

followed by a non-linear activation

$$a = f(z) \qquad \ldots 33$$

For a layer, the output is

$$a = f(Wx + b) \qquad \ldots 34$$

The network is trained by minimizing a loss function L and updating parameters using backpropagation and gradient descent (Goodfellow et al., 2016).

Ferreira et al. (2018) used this approach, the researchers aimed to detect phishing traits from websites. The result from their experiment shows that the ANN model achieved 87.61% accuracy when categorizing the phishing websites from a dataset of 1000 records from the Machine Learning and Intelligent Systems Learning Center at the University of California. The researchers further compared the ANN method with an evolving neural network that is based on reinforcement learning, the compared methods however, achieved an accuracy slightly lesser than the proposed ANN, with a difference of just 0.40%. The study thus suggested that the slight change in accuracy is a result of a change in the order of attributes. Thus, ANN performance can improve with the right order of attributes

Jasim & George (2023) also applied the ANN algorithm to the detection of phishing emails. The method used for the implementation of the research has four main phases which are feature extraction, processing, feature selection, and classification. The researcher used the k-means algorithm for the feature selection and applied the ANN for the classification. The result from the experiment shows that the ANN model was able to achieve an accuracy of 99.4%. Shoaib et al. (2023) also use the ANN algorithm for the detection of phishing URLs. The research aim is to show how different ML algorithms can perform in detecting phishing attacks. Algorithms such as NB, SVM, KNN, RF, and ANN were used in the research. The experimental result shows that the ANN model performs the best by achieving the highest accuracy of 84.84%.

Table 2: Summary of Machine Learning Algorithms for Phishing Attack Detection

Algorithm	**Dataset(s) Used**	**Accuracy (%)**	**Strengths**	**Weaknesses**
Decision Trees	PhishTank, UCI Phishing Dataset	92.5	Simple to implement, interpretable, fast on small datasets	Prone to overfitting, sensitive to noisy data
Random Forest	UCI Phishing Dataset, Kaggle	95.2	Robust to overfitting, handles missing data well	Computationally expensive with large datasets
Support Vector Machines (SVM)	Private Dataset, UCI Dataset	88.7	Effective in high-dimensional spaces, robust to overfitting	Slow training with large datasets, memory-intensive
Naive Bayes	UCI Phishing Dataset	85.4	Fast, works well with small datasets	Assumes feature independence, lower accuracy
K-Nearest Neighbors (KNN)	Custom Dataset, UCI Dataset	87.3	Simple to understand, effective for small datasets	Slow prediction time, sensitive to irrelevant features
Neural Networks (ANN)	Custom Phishing Dataset	97	Excellent for complex patterns, high accuracy	Requires large datasets, computationally expensive
Gradient Boosting (XGBoost)	UCI Dataset, PhishTank	96.3	High performance, handles imbalanced datasets well	Prone to overfitting if not tuned properly
Logistic Regression	UCI Phishing Dataset	82.1	Easy to implement, interpretable coefficients	Limited to linear relationships, lower accuracy
Ensemble Methods	UCI Dataset, Custom Dataset	96.5	Combines strengths of multiple algorithms	Computationally intensive, complex to tune

5. Comparative Analysis of Machine Learning in Phishing Attack Detection

To choose the best ML algorithm for phishing detection, many factors need to be considered. Different factors can affect the performance of the models positively or negatively. Such factors can include the, accuracy, complexity and size of the data, processing speed, and many more. Figure 2 highlights the accuracy comparison of classification algorithms, while Table 3 presents a detailed evaluation of machine learning methods, showcasing their strengths and weaknesses in phishing detection tasks for effective model selection.

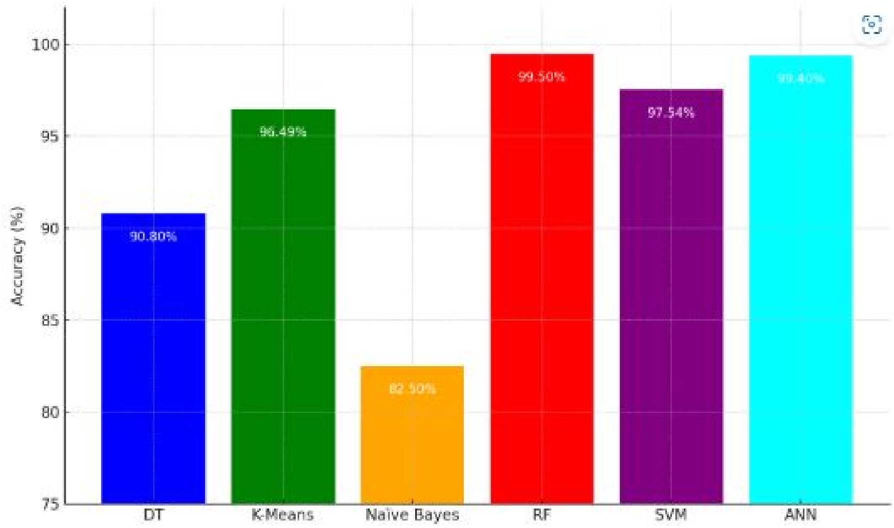

Figure 2: Comparison of classification algorithm's accuracy

Table 3: Comparative analysis of ML methods

Method	Advantages	Disadvantages
Decision Tree Algorithm	Simplicity in explaining and interpreting the feature relationships (Ashar& Maryam, 2023). DT produces easy-to-understand IF-THEN statements (Ashar& Maryam, 2023). Implementation is easy compared to others (Deepak & Nikhil, 2024). DT takes less time for classification than others (Somesha et al.,2020)	DT does not support online learning, so the tree must be rebuilt with new data, which is time-consuming(Alnemari&Alshammari, 2023). DT haslower classification results thanother ML methods (Vaitkevicius&Marcinkevicius, 2020). DT becomes more complex as featuresincrease in number (Yang et al., 2018). DT cannot deal with missing values (Pérez et al., 2023).
k-means clustering	Ability to minimize clustering error in feature space (Li et al., 2021). Easy to identify phishing patterns by clustering similar URLs (Wang & Zhou, 2020)	Results depend on initial random assignments, which can lead to poor performance if the initialization is not done correctly (Alnemari&Alshammari, 2023). Unable to classify uncertain and missing values (Pattanaik et al., 2020) Requires high computational resources and

		memory to achieve good accuracy (Pattanaik et al., 2020)
Naïve Bayes Algorithm	Simple to converge and straightforward (Verma et al., 2021). During the classification process, NB used a small amount of data to estimate important features (Verma et al., 2021). Less classification time (Sahoo et al., 2020). Ability to handle missing values by estimating probabilities for them (Sahoo et al., 2020).	Cannot learn feature relationships, leading to lower accuracy compared to another algorithm (Zhao et al., 2022). Needs a large dataset for better accuracy (Zhao et al., 2022). Requires a lot of storage space for all training samples (Verma et al., 2021). Doesn’t show variable relationships properly (Verma et al., 2021).
Random Forest Algorithm	Works efficiently on large amounts of datasets with lots of features (Chiew et al., 2019). Provides high accuracy, even on complex problems (Chiew et al., 2019). Avoids overfitting by using many trees (Kapan &Gunal, 2023). Easy to interpret results (Kapan &Gunal, 2023)	Large number of trees can slow down real-time predictions (Alnemari&Alshammari, 2023). Only predicts; and does not explain data relationships, making it hard to interpret (Alnemari&Alshammari, 2023). Sensitive to parameter changes (Kapan &Gunal, 2023). Results may vary due to random factors (Kapan &Gunal, 2023).
Support Vector Machine Algorithm	High classification accuracy (Dong et al., 2020). works well on high-dimensional data (Dong et al., 2020). Efficient with memory and converges quickly (Kumari et al., 2021). Robust in maximizing margin for prediction (Kumari et al., 2021).	Requires specific kernel settings, making it time-consuming (Alnemari&Alshammari, 2023). Difficult to interpret results (Alnemari&Alshammari, 2023). Hard to handle numerical variables (Kumari et al., 2021). Limited to binary classification (Kumari et al., 2021).

ANN	ANN can allow for specifying attributes and the type of learning used in the model (Ferreira et al., 2018). ANN is fault-tolerant and can be used on noisy or incomplete data (Thike et al., 2020). ANNcan create accurate models using experimental data (Jasim & George 2023). ANN has distributed memory, which allows it to work well in parallel processing (Yang et al., 2018)	The classification results can be affected by the data attribute order (Mridha et al., 2021). ANNs always have a slow learning rate when using a low learning rate, and a high learning rate can lead to instability (Yang et al., 2018). It is difficult to set up the problem for ANN (Hassan &Fakharudin, 2023). The result produced by ANN isn't easy to understand because ANN does not reveal the model structure (Salloum et al., 2021).

6. Discussion of the Result

The results from this research show the strengths and limitations of the different machine learning algorithms used to detect phishing attacks. DT algorithms (ID3 and C4.5), show simplicity and effectiveness, the C4.5 displayed its ability to handle large and noisy datasets. K-Means clustering when used by Al-Sabbagh et al. (2024) achieves a high accuracy of 89.2% on small datasets. NB proves efficient in text and email phishing classification, achieving over 80% accuracy in studies like Singh (2019). RF proves to be most effective by achieving up to 99.5% accuracy when used by Somesha&Pais(2022). SVM also performs well, with hybrid models reaching 97.54% accuracy when used by Elsheh&Swayeb(2023). Lastly, ANN shows potential, achieving 87.61% accuracy (Ferreira et al., 2018), with improvements dependent on feature selection. These findings underscore the importance of algorithm selection based on data characteristics and use cases.

7. Conclusion

Phishing detection is a complex challenge because of the way attackers explore human vulnerabilities and not the system error. Phishing detection is classified as a classification problem, and machine learning offers a powerful solution. It can build a predictive model that can detect phishing attempts with increased accuracy and precision. This research discussed the four features that can be considered in the detection of phishing attacks, which included URL-based, domain-based, page-based, and content-based features. This paper also looked into the major used ML algorithms in phishing detection. These algorithms include DT, RF, SVM, NB, and ANN. An in-depth comparative analysis of these ML algorithms was done including the feature optimization and mathematics representations, examining their strength and weaknessesand the overall performance of each model in phishing detection.For instance, SVM was able to achieve 97.54% accuracy in phishing link detection, this is due to its high classification accuracy when used on high-dimensional data when used by. However, its difficulty in handling numerical variables and interpreting results is due to its requirement for specific kernel settings, making it time-consuming. The DT implementation is easy and it requires less time for classification as compared to the other algorithms, but has a challenge with low classification results and becomes more complex as features increase in number . The k means clustering can easily identify phishing patterns by clustering similar URLs, but requires high computational resources and memory to achieve good accuracy. NB can use a small amount of data to estimate important features, has less classification time, and can handle missing values by estimating probabilities for them, but cannot learn feature relationships, leading to lower accuracy compared to another algorithm. RF can efficiently work on large amounts of datasets with lots of features and can give high accuracy, even on complex problems, but its large number of trees can slow down real-time predictions. ANN is fault-tolerant and can be used on noisy or incomplete data and create accurate models using experimental data, but its classification results can be affected by the data attribute order.

Our findings show that it is difficult to determine the best algorithm for phishing detection since each method has its unique advantages and disadvantages as shown in Table 3. Selecting an algorithm depends on the problem and selected features because there is no single algorithm that performs best on every problem and can be applied to different problem domains. Future research can be done on the investigation of the application of hybrid models and ensemble models on phishing detection to enhance accuracy. Ultimately, the findings contribute to the ongoing effort to fortify cybersecurity by enhancing the reliability and robustness of phishing detection systems.

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

Option 2:

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc. have been used during the writing or editing of manuscripts. This explanation will include the name, version, model, and source of the generative AI technology and as well as all input prompts provided to the generative AI technology

Details of the AI usage are given below:

1.

2.

3.

References

Adane, K., &Beyene, B. (2022). Machine learning and deep learning based phishing websites detection: the current gaps and next directions. *Review of Computer Engineering Research*, *9*(1). https://doi.org/10.18488/76.v9i1.2983

Almousa, M., Zhang, T., Sarrafzadeh, A., & Anwar, M. (2022). Phishing website detection: How effective are deep learning□based models and hyperparameter optimization? .*SECURITY AND PRIVACY*, *5*(6). https://doi.org/10.1002/spy2.256

Alnemari, S., &Alshammari, M. (2023). Detecting phishing domains using machine learning. *Applied Sciences*, *13*(8), 4649. https://doi.org/10.3390/app13084649

Al-Sabbagh, A., Hamze, K., Khan, S., &Elkhodr, M. (2024). An Enhanced K-Means Clustering Algorithm for Phishing Attack Detections. *Electronics*, *13*(18), 3677. https://doi.org/10.3390/electronics13183677

Arab, M., &Sohrabi, M. K. (2017). Proposing a new clustering method to detect phishing websites. *Turkish Journal of Electrical Engineering and Computer Sciences*, *25*(6). https://doi.org/10.3906/elk-1612-279

Ashar, A., & Maryam, D. (2023). Detecting Phishing Websites using Decision Trees: A Machine Learning Approach. *International Journal for Electronic Crime Investigation*, *7*(2). https://doi.org/10.54692/ijeci.2023.0702155

Azzani, I. K., AdiPurwantoro, S., &ZakkyAlmubarok, H. (2024). Enhancing awareness of cyber crime: a crucial element in confronting the challenges of hybrid warfare In Indonesia. *Defense and Security Studies*, *5*. https://doi.org/10.37868/dss.v5.id255

Charbuty, B., &Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01). https://doi.org/10.38094/jastt20165

Chiew, K. L., Lai, Y. H., & Tan, C. L. (2019). A hybrid feature selection strategy for phishing website detection. *Computers & Security, 83*, 256–265. https://doi.org/10.1016/j.cose.2019.03.008

Deepak, C. P., & Nikhil, C. Y. (2024). Phishing URL Detection using Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*. https://doi.org/10.48175/ijarsct-15248

Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. In *IEEE Access* (Vol. 10). https://doi.org/10.1109/ACCESS.2022.3151903

Dong, B., Liu, Y., & Zhao, Q. (2020). SVM-based phishing detection system with optimized kernel functions. *Cybersecurity Journal, 8*(4), 190–202.

Elsheh, M. M., &Swayeb, K. (2023). Phishing Website Detection Using a Hybrid Approach Based on Support Vector Machine and Ant Colony Optimization. *Proceeding - 2023*

IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2023. https://doi.org/10.1109/MI-STA57575.2023.10169464

Faris, M., Mahmud, M. N., Salleh, M. F. M., &Alnoor, A. (2023). Wireless sensor network security: A recent review based on state-of-the-art works. In *International Journal of Engineering Business Management* (Vol. 15). https://doi.org/10.1177/18479790231157220

Ferreira, R. P., Martiniano, A., Napolitano, D., Romero, M., de Oliveira Gatto, D. D., Farias, E. B. P., & Sassi, R. J. (2018). Artificial Neural Network for Websites Classification with Phishing Characteristics. *Social Networking*, *07*(02). https://doi.org/10.4236/sn.2018.72008

Ganesan, S. (2022). Detection of Phishing Websites Using Classification Algorithms. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 73). https://doi.org/10.1007/978-981-16-3961-6_12

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Available online: https://www.deeplearningbook.org

Guaña-Moya, J., Chiluisa-Chiluisa, M. A., Jaramillo-Flores, P. del C., Naranjo-Villota, D., Mora-Zambrano, E. R., &Larrea-Torres, L. G. (2022). Phishing attacks and how to prevent them. *Iberian Conference on Information Systems and Technologies, CISTI*, *2022-June*. https://doi.org/10.23919/CISTI54924.2022.9820161

Harris, S., & de Amorim, R. C. (2022). An Extensive Empirical Comparison of k-means Initialization Algorithms. *IEEE Access*, *10*. https://doi.org/10.1109/ACCESS.2022.3179803

Hassan, N. H., &Fakharudin, A. S. (2023). Web Phishing Classification Model using Artificial Neural Network and Deep Learning Neural Network. *International Journal of Advanced Computer Science and Applications*, *14*(7). https://doi.org/10.14569/IJACSA.2023.0140759

Jagadeesan, S., Chaturvedi, A., & Kumar. S. (2018). URL Phishing Analysis using Random Forest. *International Journal of Pure and Applied Mathematics*, *118*(20).

Jasim, M. & George, L. E. (2023). Phishing Attacks Detection by Using Artificial Neural Networks. *Iraqi Journal for Computer Science and Mathematics*. https://doi.org/10.52866/ijcsm.2023.02.03.013

Jupin, J. A., Sutikno, T., Ismail, M. A., Mohamad, M. S., Kasim, S., &Stiawan, D. (2019). Review of the machine learning methods in the classification of phishing attack. *Bulletin of Electrical Engineering and Informatics*, *8*(4). https://doi.org/10.11591/eei.v8i4.1344

Kalhor, A. (2020). An introduction to artificial neural networks. In *Hardware Architectures for Deep Learning*. https://doi.org/10.1049/PBCS055E_ch1

Kapan, S., &Gunal, E. S. (2023). Improved phishing attack detection with machine learning: A comprehensive evaluation of classifiers and features. *Applied Sciences*, *13*(24), 13269. https://doi.org/10.3390/app132413269

Krishna, Mr. B. (2021). E-Mail Spam Classification using Naive Bayesian Classifier. *International Journal for Research in Applied Science and Engineering Technology*, *9*(VI), 5209–5214. https://doi.org/10.22214/ijraset.2021.36153

Kumari, S., Gupta, D., & Rana, S. (2021). Support vector machine performance in phishing detection: A survey. *Journal of Computer Applications*, *18*(6), 43–50.

Li, Y., Wang, J., & Lin, Z. (2021). Clustering and classification-based phishing detection methods: A review. *IEEE Access*, *9*, 76247–76261. https://doi.org/10.1109/ACCESS.2021.3054549

Mitra, S., & Padmanabhan, D. (2023). A survey of decision trees: Concepts, algorithms, and applications. *IEEE Transactions on Knowledge and Data Engineering*.

Mridha, K., Hasan, J., Saravanan, D., & Ghosh, A. (2021). Phishing URL Classification Analysis Using ANN Algorithm. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies, GUCON 2021*. https://doi.org/10.1109/GUCON50781.2021.9573797

Nakhipova, V., Kerimbekov, Y., Umarova, Z., Suleimenova, L., Botayeva, S., Ibashova, A., &Zhumatayev, N. (2024). Use of the Naive Bayes Classifier Algorithm in Machine Learning for Student Performance Prediction. *International Journal of Information and Education Technology*, *14*(1). https://doi.org/10.18178/ijiet.2024.14.1.2028

Pajila, P. J. B., Sheena, B. G., Gayathri, A., Aswini, J., Nalini, M., & Siva Subramanian, R. (2023). A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications. *Proceedings of the 4th International Conference on Smart Electronics and Communication, ICOSEC 2023*. https://doi.org/10.1109/ICOSEC58147.2023.10276274

Pattanaik, S., Panigrahi, R., &Samal, B. (2020). Enhancing k-means performance for phishing detection in IoT environments. *Future Internet*, *12*(5), 86. https://doi.org/10.3390/fi12050086

Pérez, A., Sandoval, L., &GarcíaVillalba, L. J. (2023). Analysis of Machine Learning Techniques for Information Classification in Mobile Applications. In *Applied Sciences (Switzerland)* (Vol. 13, Issue 9). MDPI. https://doi.org/10.3390/app13095438

Pirouz, B., &Pirouz, B. (2023). Multi-Objective Models for Sparse Optimization in Linear Support Vector Machine Classification. *Mathematics*, *11*(17). https://doi.org/10.3390/math11173721

Rajoju, R., Sathvika, V., Smaran, G. N. S., Tejashwini, C., & Reddy, G. A. (2024). Text Phishing Detection System using Random Forest Algorithm. *Proceedings of the 3rd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2024*. https://doi.org/10.1109/ICAAIC60222.2024.10575110

Sahoo, J., Kaur, A., & Malhotra, P. (2020). Probabilistic approaches in phishing detection systems: A Naive Bayes perspective. *International Cybersecurity Journal*, *25*(2), 88–100.

Saini, A. (2024). Guide on Support Vector Machine (SVM) Algorithm. *Analytics Vidhya*.

Salloum, S., Gaber, T., Vadera, S., &Shaalan, K. (2021). Phishing Website Detection from URLs Using Classical Machine Learning ANN Model. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *399 LNICST*. https://doi.org/10.1007/978-3-030-90022-9_28

Sankhyan, R., Shetty, A., Dhanopia, L., Kaspale, C., &Dantal, G. (2018). PDS-Phishing Detection Systems. *International Research Journal of Engineering and Technology*, *5*(4).

Saputra, M. F. A., Widiyaningtyas, T., &Wibawa, A. P. (2018). Illiteracy classification using K means-naïve bayes algorithm. *International Journal on Informatics Visualization*, *2*(3), 153–158. https://doi.org/10.30630/joiv.2.3.129

Saraswathi, P., Anchitaalagammai, J. v., & Kavitha, R. (2023). A System Review on Fraudulent Website Detection Using Machine Learning Technique. *SN Computer Science*, *4*(6). https://doi.org/10.1007/s42979-023-02084-6

Shantanu, J. B., & Joshua A. K. R. (2021). Malicious URL Detection: A Comparative Study. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*. https://doi.org/10.1109/ICAIS50930.2021.9396014

Shoaib, M., & Umar, M. S. (2023). Comparative Analysis using Machine Learning Techniques for Detecting and Mitigating Phishing. *2023 2nd International Conference on Smart Technologies for Smart Nation, SmartTechCon 2023*. https://doi.org/10.1109/SmartTechCon57526.2023.10391319

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.2988796

Singh, M. (2019). Classification of spam email using intelligent water drops algorithm with Naïve bayes classifier. *Advances in Intelligent Systems and Computing*, *714*. https://doi.org/10.1007/978-981-13-0224-4_13

Somesha, M., &Pais, A. R. (2022). Classification of Phishing Email Using Word Embedding and Machine Learning Techniques. *Journal of Cyber Security and Mobility*, *11*(3), 279–320. https://doi.org/10.13052/jcsm2245-1439.1131

Somesha, M., Pais, A. R., Rao, R. S., &Rathour, V. S. (2020). Efficient deep learning techniques for the detection of phishing websites. *Sadhana - Academy Proceedings in Engineering Sciences*, *45*(1). https://doi.org/10.1007/s12046-020-01392-4

Team, G. L. (2023). Random forest Algorithm in Machine learning: An Overview. *Great Learning Blog: Free Resources What Matters to Shape Your Career!*

Thakur, K., Ali, M. L., Obaidat, M. A., &Kamruzzaman, A. (2023). A Systematic Review on Deep-Learning-Based Phishing Email Detection. In *Electronics (Switzerland)* (Vol. 12, Issue 21). https://doi.org/10.3390/electronics12214545

Thike, P. H., Zhao, Z., Shi, P., &Jin, Y. (2020). Significance of artificial neural network analytical models in materials' performance prediction. In *Bulletin of Materials Science* (Vol. 43, Issue 1). https://doi.org/10.1007/s12034-020-02154-y

Vaitkevicius, P., &Marcinkevicius, V. (2020). Comparison of Classification Algorithms for Detection of Phishing Websites. *Informatica (Netherlands)*, *31*(1). https://doi.org/10.15388/20-INFOR404

Verma, A., Gupta, S., & Kumar, P. (2021). Naïve Bayes classifiers for phishing detection: Recent advances and challenges. *International Journal of Machine Learning Applications, 14*(3), 98–115.

Wang, L., & Zhou, T. (2020). A feature engineering approach to phishing detection using k-means. *Journal of Data Science, 18*(9), 119–132.

Wang, S., Li, M., Hu, N., Zhu, E., Hu, J., Liu, X., & Yin, J. (2019). K-Means Clustering With Incomplete Data. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2910287

Wood, T., Basto-Fernandes, V., Boiten, E., &Yevseyeva, I. (2022). Systematic Literature Review: Anti-Phishing Defences and Their Application to Before-the-click Phishing Email Detection. *ArXiv*.

Wu, J., Lin, X., Yang, X., Li, S., Zhang, B., & Gao, L. (2023). Research on random forest algorithm based on oversampling and feature selection. *ITOEC 2023 - IEEE 7th Information Technology and Mechatronics Engineering Conference*. https://doi.org/10.1109/ITOEC57671.2023.10291457

Yang, S., Feng, Q., Liang, T., Liu, B., Zhang, W., &Xie, H. (2018). Modeling grassland above-ground biomass based on artificial neural network and remote sensing in the Three-River Headwaters Region. *Remote Sensing of Environment, 204*. https://doi.org/10.1016/j.rse.2017.10.011

Zhao, X., Li, J., & Wen, H. (2022). Naïve Bayes methods for email phishing detection: Challenges and future directions. *Computers & Security, 116*, 102527. https://doi.org/10.1016/j.cose.2022.102527

Zouina, M., &Outtaj, B. (2018). A novel lightweight URL phishing detection system using SVM and similarity index. *Human-Centric Computing and Information Sciences*, *7*(1). https://doi.org/10.1186/s13673-017-0098-1