Al-Khwarizmi Engineering Journal
ISSN (printed): 1818 – 1171, ISSN (online): 2312 – 0789
Vol. 21, No. 2, June, (2025), pp. 117–135

Al-Khwarizmi Engineering Journal

K-Gen PhishGuard: an Ensemble Approach for Phishing Detection with K-Means and Genetic Algorithm

Ali Raheem Al-Hafiz[1*], Adnan J. Jabir[2] and Shamala Subramaniam[3]
[1, 2] *Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*
[3] *Department of Communication Technology and Networking, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, 43400, Selangor Darul Ehsan, Malaysia*
*Corresponding Author's Email: aliraheem2201m@sc.ubaghdad.edu.iq

(Received 17 November 2024; Revised 9 March 2025; Accepted 20 April 2025; Published 1 June 2025)
https://doi.org/10.22153/kej.2025.04.011

Abstract

Phishing detection is considered a critical problem in cybersecurity, and utilising machine learning with an efficient feature selection method for precisely identifying malicious websites is deemed the most critical challenge. This research presents a two-phase phishing detection system by employing unsupervised feature selection and supervised classification. In the first phase, the best set of features is identified by the Genetic algorithm and is utilised by the K-means clustering algorithm to divide the dataset into groups with similar traits. In the second phase, the best set of features in each group is identified through the Genetic algorithm to enhance the classification process. Finally, a voting ensemble technique is applied, in which the Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting (XGBoost) and Adaptive boosting (AdaBoost) models are combined. Predictions are aggregated using a soft voting mechanism. This research utilises the web page phishing detection dataset, which consists of 11,430 URLs with 87 features. From the results, an accuracy of 99% is achieved using the voting ensemble technique with feature selection compared with 77.3% without feature selection. The model performance experiences a significant boost through the GA-optimised feature selection by reducing computational complexity and improving key metrics such as accuracy, precision and F1-score. Additionally, the performance across four clusters demonstrates the positive impact of K-Means clustering in improving classification accuracy for specific data groups. As proven by the obtained results, integrating feature selection with ensemble learning is effective for phishing detection; moreover, the scalability and efficiency of such a solution in real-world applications are demonstrated.

Keywords: *AdaBoost; ensemble learning; feature selection; genetic algorithm; K-means clustering; machine learning; phishing detection*

1. Introduction

Phishing is a popular cyber threat, where the attacker tricks the users to disclose sensitive information, including credit card numbers, usernames and passwords. They create fake websites that resemble legitimate ones. The frequency of phishing attacks has seen a dramatic rise with the rapid increase in Internet usage, now reaching nearly 1.13 billion websites and 5.3 billion users globally as of October 2023 [1]. The COVID-19 pandemic doubled down on this surge, cementing the rise of online platforms as a way to work, learn and bank. This shift has been exploited by cybercriminals, using fake emails as phishing tools to create major losses in customer trust, reputational damage to businesses and financial consequences [2, 3].

The term phishing attacks refers to a type of social engineering in which attackers exploit the trust that the natural person has towards the attacker and thus not a technical vulnerability. Phishing is especially dangerous because it can bypass the most sophisticated cybersecurity systems. Social engineering attacks amounted to $121.22 billion in damages to the U.S. in 2016,

followed by similar impacts globally, as the U.S. Department of Justice noted [4].

However, phishing URL detection in web data is a difficult task because web data are generally unstructured and unpredictable. Researchers have adopted feature selection techniques to improve classifier performance by identifying the most influential features to enhance phishing detection. These techniques are critical in reducing model complexity to preserve or upgrade its detection ability [5]. Recent works have demonstrated that the effectiveness of Principal Component Analysis (PCA) in retaining important patterns of URL or HTML features enhances the performance of machine learning based phishing detector models [6]. High dimensional data with many relevant or redundant features, constitute a major hindrance of traditional machine learning models for phishing detection. It decreases accuracy and detection speed whilst increasing the computational costs. The research therefore proposed a two-phase system for phishing detection, where the first phase contains a hybrid model combining K-means clustering and Genetic Algorithm to determine the best feature set used for producing highly separated clusters. Reducing complexity in datasets by grouping similar data points is achieved through k-means clustering, whilst the GA improved feature selection by identifying the most relevant features using genetic algorithm. In the second stage, a system using a genetic algorithm with a voting ensemble model is proposed which enhances the system's capacity to differentiate between phishing and legitimate sites, leading to improved detection accuracy and decreased false positives and false negatives.

The key contributions of this research include the following:

• Enhanced Feature Selection: The combination of Genetic algorithm and K-means contributes to determining the best feature set to separate the dataset in different groups.

• Select Classifier: The genetic algorithm was employed to select the classifier in the ensemble learning process based on the performance metrics results.

• Improved Accuracy: By selecting the most critical features, the model increases its ability to accurately detect phishing websites.

2. Related Work

Researchers have developed various methods to fight against0 phishing attacks in the ongoing battle. However, web browsers, such as Google Chrome, Microsoft Edge and Firefox, first used whitelisting and blacklisting method to identify phishing sites whilst allowing legitimate sites and blocking malicious ones [7]. Google even took this one step further and offered a Blacklist API which acts as a database of unsafe websites and IP addresses, assisting users in verifying URLs. However, these early methods had huge drawbacks. They were susceptible to minor URL changes and zero day phishing attacks and were prone to creating a high false positive rate [7]. In light of these limitations, other sophisticated techniques started to appear. Features such as CSS files were used to differentiate phishing sites from legitimate ones and serve as a pivotal approach in Visual similarity detection [8]. The Link Guard algorithm [9] enhanced this approach by analysing the visual similarities between target pages and known phishing sites through an image-based matching method. To further improve detection accuracy and address the weaknesses of purely visual methods, researchers began to integrate visual similarity with machine learning techniques [10].

Additionally, a line of defense heuristic-based methods have emerged to analyse and determine the legitimacy of different website features. Such methods have been categorised as content based, where the content of a website and the host information or URL of non-content based ones are examined [11-14]. Some notable techniques included Phish Net [15] in which websites are classified using URL heuristics and FSM [7] that monitors web form usage. Although these heuristics were less prone to false positives than list-based approaches, they often remained inaccurate and could be challenged by attackers who know the system. The advent of machine learning (ML) significantly transformed phishing detection [16, 17]. ML algorithms trained on datasets that analyse web addresses, site structures and code can automatically detect phishing sites [18]. Early ML models, such as Artificial Neural Networks (ANN), achieved an accuracy rate of 83.38% [19]. As researchers developed methods to handle larger and more complex datasets, ML model accuracy surged beyond 99% [20]. Deep learning (DL) techniques further advanced this field; for instance, a model employing Deep Neural Networks (DNN) and Support Vector Machines (SVM) achieved 96% accuracy by training on a dataset with 28 features [21]. Another innovative approach combined Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for rapid

URL-based detection, reaching an accuracy of 97% [22].

Nonetheless, hybrid techniques, which combine machine learning with deep learning, have proven to be very successful. To illustrate, the accuracies of up to 97.98% were achieved with Random Forest (RF) and Natural Language Processing (NLP) with feature selection; in addition, combinations of methods incorporating RF with Neural Networks (NN) and K-Nearest Neighbors (KNN) showed an accuracy between 97.2% and 97.4%, respectively [23]. Furthermore, high accuracies produced by ANN and RF coupled approaches were comparable, with accuracies of 96.36%–97.33%. In some advanced models which included classifiers such as RF, SVM, Generalized Linear Models (GLM) and Generalized Additive Models (GAM), accuracy of 98.34% was achieved for a single case. These hybrid methods exemplify the growing effectiveness of machine learning and deep learning in accurately detecting phishing attacks.

The machine learning models used for phishing detection have received a recent research deal to apply Genetic Algorithms (GAs) for feature selection and hyper parameter optimisation. GA-based feature selection in one study resulted in a more accurate model (95.34% on test set). This research demonstrates the effectiveness of GAs as a highly efficient method for selecting features and model performance in ways that are simultaneously simple and expressive [24]. More recently, emerging phishing detection models [25] have also enhanced accuracy with the combined use of feature selection, clustering and classification techniques. Another approach combines the components of Rough Set (RS), Stability-Correlation (SC) and Principal Component Analysis (PCA) for feature selection; then, it fits the data to phishing and non-phishing categories through K-Means clustering. The model employed Deep Learning (DL) and Decision Tree (DT) algorithms for classification, yielding the following impressive results: The accuracy for the Decision Tree model was 97.2%, whilst the Deep Learning model reached 96.88%. The outcomes presented here demonstrate the effectiveness of hybrid methodologies in improving phishing detection over traditional techniques.

However, other studies have focused on utilising unsupervised learning for detecting phishing websites; K-Means is a widely used unsupervised Machine Learning algorithm that should be used to detect the webpage of phishing sites [26]. The algorithm clusters the websites into the following two groups: set up to distinguish between the legitimate SRI websites and phishing websites based on ‘features’ extracted from the URLs of the websites [27]. Some features include URL length, domain age and IP addresses in URL where K-Means is suggested to be implemented by researchers for identifying phishing sites. The algorithm can successfully classify the sites between the phishing and the authentic ones, which will help in the detection of the threats.

In one research [28], the author designed a model of a K-Nearest Neighbors algorithm integrated with Support Vector Machine to classify a site as phishing, legitimate or suspicious; the suggested model maintained over 98% accuracy. A major strength of the K-Means algorithm in anti-phishing lies in its simplicity, scalability and effectiveness at working with noisy data.

Recently, GA-based feature selection has been employed to identify optimal subsets of features, reducing the dimensionality and computational workload of models [7]. With this GA-driven method, high accuracy and recall levels were observed across classifiers, such as Random Forest (92.93% accuracy and 89.05% recall) and XGBoost (91.53% accuracy and 87.24% recall). The potential of GAs to enhance URL-based phishing detection has been demonstrated, contributing to research on evolutionary algorithms for improving cybersecurity accuracy and robustness.

3. Background

Machine learning has many ways to handle data challenges, and no single algorithm will solve every problem. Specific task, data involved and the model that best fits the situation determines the choice of algorithm [29]. In this thesis, five well-known machine algorithms, namely support vector machine random forest, adaptive boosting, extreme gradient boosting and ensemble learning algorithms are adopted.

3.1. Support Vector Machine

This algorithm belongs to one of the most ubiquitous in the field of machine learning applicable to classification and regression goals. It determines the best hyperplane, which separates the data into different classes or estimates the target variable in the case of regression. The basic concept of SVM lies in determining the hyperplane when the distance of the two nearest points belonging to two different classes is at its maximum. This margin is the distance from the

hyperplane to the closest points from each class in the separated space. SVM attempts to search for a hyperplane that can be used to classify the classes whilst providing the maximum margin to gain the best generalisation [30]. For non-linearly separable data, SVM has kernel trick in which the data are transformed to random hyperspace to make it easily separable. SVM can then attempt to look for a hyperplane which will split the data in question into different classes [31].

3.2. Random Forest

The random forest algorithm is used for regression and classification tasks. It belongs to the family of ensemble learning methods, which combine multiple models to improve the predictive performance of a single model [24]. In random forest, multiple decision trees are constructed independently, each based on a randomly selected subset of the training data and a random subset of the features. The output of the random forest model is then the average or the majority vote of the outputs of the individual trees [32].

3.3. Adaptive Boosting

Adaptive Boosting (AdaBoost) is a popular machine learning algorithm introduced in 1995 by Yoav Freund and Robert Schapiro. It is inspired by the idea of a gambler in horse racing seeking advice from experienced bettors to select the best horse. The primary goal of AdaBoost is to enhance the performance of multiple 'weak learners' by combining them into a single 'strong learner' capable of accurate predictions. Implementing AdaBoost is easy, and it can be used on a large variety of problems. It shows high performance, working well with low noise data whilst achieving high accuracy. The algorithm is also flexible, because it can be combined with other weak learners, such as decision trees. It also works well with errors by focusing on misclassified samples and improving the model's overall accuracy in an iterative manner [33].

3.4. Extreme Gradient Boosting

XGBoost is a scalable tree boosting with efficiency and memory resource. It is applicable for regression and classification problems. At each step, it creates a weak learner and adds to the overall model. Gradient Boosting Machines (GBMs) are created when the weak learner for each step is the gradient direction of the loss function [34].

3.5. Ensemble Learning

Ensemble learning is a machine learning technique in which the predictions of multiple models, known as 'base learners' or 'weak learners', are combined to improve overall performance. The underlying principle is that more accurate and robust predictions can be produced by a group of diverse models working together than by any single model. This approach is highly effective in various problem domains, including artificial intelligence, machine learning, pattern recognition and data mining; additionally, it has numerous real-world applications. Ensemble learning methods rely on the diversity of base models, where different models may make different errors. By combining them, the weaknesses of individual models can be offset. The predictions of these models are aggregated using techniques such as voting (for classification), averaging (for regression) or weighted combinations [35].

3.6. Feature Selection

Feature selection is a preprocessing technique aimed at identifying a minimal subset of features that effectively capture the relevant properties of a dataset for adequate classification by removing irrelevant and redundant information; it reduces data dimensionality, enhancing the speed and effectiveness of learning algorithms. The goal is to determine a subset that performs as well as or better than the original dataset. Features, which can be discrete, continuous or nominal, are categorised as relevant, unnecessary or redundant features that do not directly impact the output and fail to add uniquely to the learning target. Irrelevant features can be defined as those without influence on the result, with their values created randomly for each case. Redundancy occurs when one feature can take on a role of another [36]. Feature selection algorithm primarily aims to identify a subset of features that are both independent and effectively relevant to the learning process. Feature selection is classified into three categories: filter, wrapper and embedded techniques [37].

3.7. Genetic Algorithm

An optimisation algorithm is based on the principles of natural selection [38]. The algorithm is population based and is implemented in keeping with the principle of natural selection [39]. New populations are generated by using genetic operators across generations on current individuals within the population. The parts of GA include initialisation, fitness function, selection, crossover, mutation, replacement and termination [40].

3.7.1. Initialisation

Initialisation produces a starting population of chromosomes or possible solutions. A possible set of characteristics for phishing website detection is represented by each chromosome.

3.7.2. Fitness Function

Fitness function refers to a measure that evaluates the quality or 'fitness' of solutions within a given space, determining how effectively they address the problem. It assigns a fitness score based on the solution's ability to meet the problem's objectives, guiding the evolutionary process. Solutions with higher fitness scores tend to be selected for reproduction and carried forward to subsequent generations [41].

3.7.3. Selection

Selection is a crucial phase of genetic algorithms because it defines which strings will be involved in reproduction. It is also known as the reproduction operator. Selection pressure influences the convergence rate of GA. Some of the methods used are roulette wheel, rank, tournament, Boltzmann and stochastic universal sampling [42].

3.7.4. Crossover

Operators are utilised to generate children by using the genetic data of two or more parents. The frequently used crossover operators are single-point crossover, two-point crossover, k-point crossover and uniform crossover. Some common types of crossover techniques used in genetic algorithms include partially matched crossover, order crossover, precedence-preserving crossover, shuffle crossover, reduced surrogate crossover and cycle crossover [43].

3.7.5. Mutation

In genetic algorithms, mutation is a key genetic operator used to maintain genetic diversity from one generation to the next. It introduces small, random changes to individual solutions (chromosomes) in the population. This step helps the algorithm avoid premature convergence to suboptimal solutions by exploring new areas in the search space [42].

3.7.6. Replacement

Replacement is the process of changing the old one with new offspring population. When the population has been updated, the cycle is repeated now [42].

3.7.7. Termination

Termination is a process of continuing to select, crossover and mute these sets until the convergence requirements (e.g., to achieve a particular fitness level or for a fixed number of generations) are satisfied [43].

3.8. K-Means

K-means is one amongst the simplest unsupervised learning algorithms used to solve clustering problems. The procedure here involves an easy and simple process of factoring a given dataset through a certain number of clusters. The concept is to determine k centres, in which one is found for each cluster of the dataset. These centres should be placed cunningly because a different location brings a different outcome. Thus, individually keeping the frequency as much as possible at a distance from others is a wise move [44].

3.8.1. Elbow Method

The earliest approach that has ever existed is commonly referred to as the Elbow method. It operates by evaluating the value or percentage of a tested variable k and establishing an elbow at a specific point. The value of k in a combination of elbows with K-Means represents a graph displaying cluster relationships with reducing errors. Increasing the k value causes the graph to decrease slowly until the k value is stable [25]. This method works by evaluating the Within-Cluster Sum of Squares (WCSS) for different values of k.

$$WCSS(k) = \sum_{j=1}^{k} \sum_{xi \in Cj} |xi - nj|, \quad \ldots(1)$$

3.8.2. Silhouette

Silhouette is a metric used to evaluate the quality of clustering within a dataset. It quantifies the degree of similarity between a data point and its own cluster (cohesion) as opposed to other clusters (separation). This metric can be used to decide on the number of clusters that should be formed [45]. The Silhouette formula is

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \quad \ldots(2)$$

4. Proposed Model

A new hybrid approach is used in the proposed model to detect phishing websites. Unsupervised and supervised learning techniques are combined to add accuracy and efficiency to the process. By combining K-means clustering with a GA, feature selection is optimised, which is considered a critical component in improving the performance of machine learning classification models. This hybrid method allows for a refined analysis of website features, with redundancy reduced and the dataset partitioned into parts based on similarity. In addition, the best feature set is selected for each resulting cluster through the GA algorithm to enhance classifier performance, thereby enabling the voting model to better differentiate between phishing and legitimate sites. The section below displays a detailed breakdown of the model's workflow represented in three distinct phases, as shown in Fig. 1.

In Phase 1, the process is initiated with the loading and preprocessing of the dataset to ensure preparedness for analysis. From this step, an initial population of feature subsets is generated. These feature subsets are subjected to unsupervised learning, and K-means clustering is applied to group dataset samples based on similarity using a chosen feature subset. The quality of the clustering is evaluated using the silhouette fitness function, allowing the performance of each feature subset to be gauged. Genetic algorithm (GA) operations—selection, crossover and mutation—are utilised to enhance the feature selection process by evolving the population over multiple iterations. This phase is concluded with the identification of the best feature subsets and cluster centres, thus setting the stage for the subsequent phase.

Phase 2 incorporates supervised learning into the process. The dataset is divided into clusters around the centres and selected features based on the results obtained from Phase 1.

An advantage of clustering is that for each cluster, a classification model is created by using advanced machine learning algorithms, such as SVM, Adaboost, Random Forest or XGBoost. The feature subset is iteratively refined by the GA, with the best-performing features selected using genetic operations and repeated looping. The models are then assessed on the basis of their accuracy, precision and F1 scores to ensure that only the most effective feature sets and models are selected. This phase concludes with another round of inspection through the system to strengthen the classification model for each cluster, thereby enhancing its ability to detect phishing threats.

Phase 3 prepares the last phase of the system for actual application in the real world. The dataset is preprocessed once again, and each sample is assigned to its appropriate cluster. Predictions are made by the trained models for corresponding clusters using the selected features. For robustness, an ensemble method was adopted, whereby the final decision was synthesised through a voting mechanism that incorporated the output of several models. The accuracy and reliability of the system are increased by this collective decision-making approach, thereby mitigating the presence of phishing websites in real time. Algorithm 1 demonstrates these three phases.

Algorithm I: K-GenPhishGuard

Input:
Dataset D
Output:
Final fishing detection ensemble models
Begin:
1. Preprocessing
- Duplicated sample removal.
- Null and constant sample removal.
- Normalisation.
- Class label encoding.
2. GenKMeans:
- The GA is utilised to determine the best features for K-Means clustering.
- The output comprises cluster centres and best feature subset.
4. GenClassification:
- The GA is utilised to determine the best features to train the ensemble set of classifiers for each cluster.
- The output is the final fishing detection ensemble models for each cluster.
End

The first phase deals with the data preprocessing process, where the dataset is loaded, labels are encoded into numerical format and features are scaled using MinMaxScaler to ensure consistent ranges across all features. The data are then prepared for the next phase. The second phase, namely GenKMeans, integrates the GA with K-means clustering to determine the best features that achieve the minimum silhouette score. The strength of K-means clustering is leveraged in identifying natural groupings within the data, and the GA's ability to evolve feature subsets is utilised to refine the model's accuracy over multiple iterations. By combining these techniques, an efficient and precise feature selection process is ensured, ultimately enhancing the performance of the phishing detection system. Algorithm 2 shows the main steps of this phase.

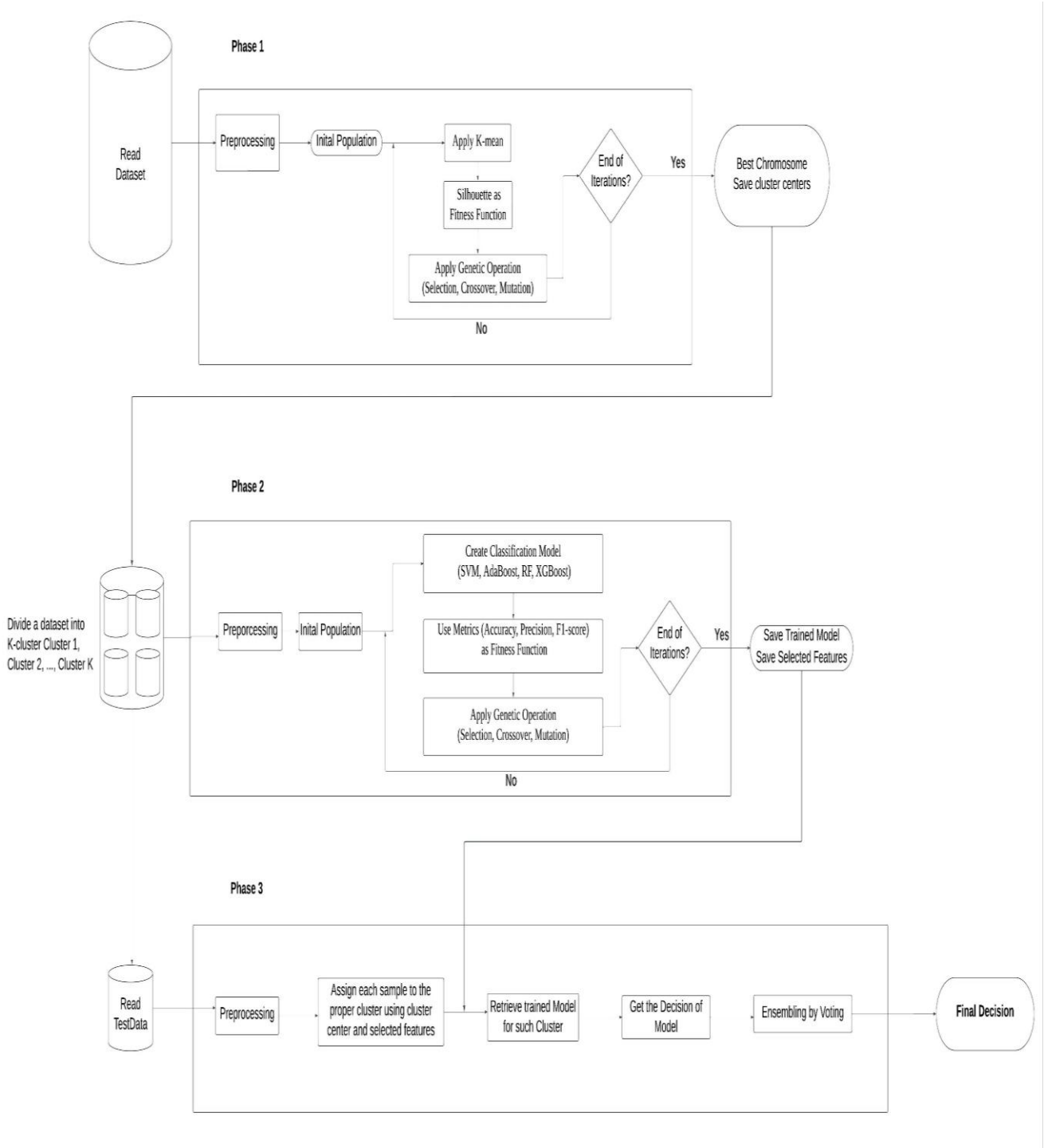

Fig. 1. Proposed K-Gen PhishGuard model

Algorithm 2: GenKMeans

Input:

Pre-processed Dataset (Dp).

The number of clusters (K) obtained using the elbow method.

Output:

• Best feature subset F_{best} {f1, f2..., fm}

• Cluster Centres = {C1, C2..., Ck}

Begin:

1. Generate initial population P = {ch1, ch2..., chp}, where each chromosome is a binary vector representing the selected features.
2. Without stopping the condition, do steps 3–4.
3. For all solutions in (P) = {ch1…chp} do
 - Perform K-Means Clustering using ch(i).
 - Use Silhouette score as the fitness function:

$$S = \frac{1}{n}\sum_{i=1}^{n}\frac{b(i)-a(i)}{\max\left(a(i),b(i)\right)}.$$

4. Perform selection, crossover and mutation.
5. Save the best features F = {f0, f1,...fm} and the cluster centres C = {C1, C2,...Ck}.

5. End

As shown in the GenKMeans, the initial binary population is generated, where each vector is represented by the selected features. Each binary value within the vector is used to indicate whether a particular feature is included in the model (1) or excluded (0). The starting point for the GA is served by this population. The process is then moved to K-Means clustering, with the latter performed on the basis of the selected features in each solution of the GA population.

The Silhouette score is used here as the fitness function, which measures how well each data point is clustered on the basis of its cohesion within its own cluster and its separation from other clusters. The GA proceeds with multiple generations of feature selection, using selection, crossover and mutation to evolve the population. The goal is to maximise the silhouette score by determining the optimal subset of features that enhances clustering performance. Once the GA has converged, the best-performing chromosome, representing the optimal set of features, is selected. In addition, the cluster centres are saved for use in the next phase. The output of the previous phase, i.e., the cluster centres and the best feature subset, is taken as input for the third phase. Then, for each cluster, the best feature subset, which achieves the best classification accuracy, is identified using the GA. Algorithm 3 shows the detailed steps of this phase.

Algorithm 3: GenClassification

Input:

- Cluster Centres C = {C0, C1,...Ck}.
- Clustering Features F = {f0, f1,...fm}

Output:

- Classification feature vectors CF = {Cf0, Cf1,...Cfm}.
- Trained classification models.

Begin

1. For each cluster, do 2–6.
2. Generate initial population P = {ch1, ch2..., chp}, where each chromosome is a binary vector which represents the selected features.
2. Without stopping the condition, do 3–5.
3. For all solutions in (P) = {ch1...chp}, do 4–5.
4. Fitness Evaluation:
   - Train ensemble classifiers (SVM, RF, XGBoost and AdaBoost) using the selected features represented by ch (i).
   - Evaluate performance using metrics such as accuracy, precision and F1 score using the voting method.
5. (GA) for Optimisation:

Selection, Crossover and Mutation operators.

6. Output:

The best feature subset CF_{best}.

The trained ensemble classification models.

End

As shown in Algorithm 3, all clusters are iterated upon in Step (1), whilst corresponding with the initialisation stage in Step (2), where a population of binary chromosomes is created during the initial population generation step. Each possible feature subset is represented by a chromosome, and a binary value is assigned to each feature: (1) if the feature is selected; otherwise, (0). Step (3) iterates all GA solutions whilst performing the fitness evaluation using a voting method in Step (4). For each chromosome, the subset of selected features is used to train the ensemble set of classifiers, which is formed by combining SVM, RF, XGBoost and AdaBoost. The classifier's performance is evaluated using key metrics such as accuracy, precision and the F1 score. These metrics are used as the fitness function to determine the quality of the feature subset.

Step (5) optimises feature selection by the GA through the following three main operations: selection, crossover and mutation. In the selection process, chromosomes are chosen on the basis of their fitness scores using the roulette wheel method. Diversity is introduced by crossover and

mutation, allowing the algorithm to explore new feature combinations. The population is updated at each iteration, and the best chromosome is retained on the basis of fitness.

Finally, the output step obtains the best feature subset and the trained ensemble models (based on the best feature selection). This step ensures that the optimal combination of features and classifiers is used for future predictions.

5. Dataset

This study uses a dataset with 11,430 URLs and 87 features extracted for phishing detection. These features are categorised into the following three primary groups: Content features (24), URL structure and syntax features (56) and external service features (7). The URL structure and syntax features capture the patterns of the construction of the URLs to identify phishing attempts. The model is trained on webpages associated with URLs. Additionally, content features are produced, which can be used to assess if phishing activity is indicated by the content. The classification process is extended to include external service features from third-party sources, such as reputation scores.

To clearly understand the results of the classifiers, the dataset and its components must first be comprehended. Therefore, a detailed summary study of the components of the dataset and their respective proportions was conducted. The dataset is balanced, with 50% phishing URLs and 50% legitimate URLs, ensuring that one class is not favoured over the others, as shown in Fig. 2. This balance is critical for unbiased classification results. The dataset, sourced from Kaggle [46], is widely recognised as a standard benchmark for phishing detection, rendering it suitable for reproducibility and comparison across studies.

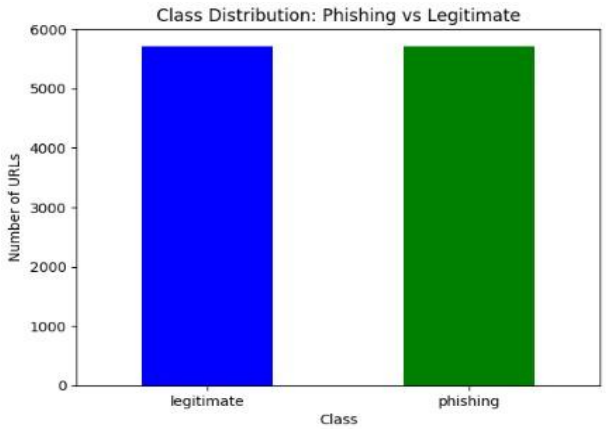

Fig. 2. Dataset class distribution for phishing

6. Metrics and Results

6.1. Metrics

The proposed work evaluates the performance of the phishing detection classifiers to ensure that accurate identification of phishing websites is achieved whilst minimising errors. Several metrics have been utilised to assess different aspects of model effectiveness, especially in handling the balanced dataset of phishing and legitimate URLs. These metrics include accuracy (Acc), which is used to measure the model's overall correctness; precision (P), which focuses on the accuracy of phishing predictions; and the F1 Score (F1), which is used to provide a balance between precision and recall for measuring the model's effectiveness in detecting phishing sites. These metrics are formulated in Eqs. (1–3), respectively.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN}, \quad \ldots(3)$$

$$P = \frac{TP}{TP + FP}, \quad \ldots(4)$$

$$F1 - Score = 2 \times \frac{R \times P}{R + P}, \quad \ldots(5)$$

where R is the recall function and can be expressed as

$$R = \frac{TP}{TP + FN}, \quad \ldots(6)$$

6.2. Result

The results presented in this paper are discussed with two scenarios. The first scenario is focused on examining three different cases within the same proposed model to demonstrate the impact of feature selection through the Genetic Algorithm (GA). This selection process enhances the accuracy of phishing detection by combining ensemble learning with the Genetic Algorithm. The second scenario involves comparing the proposed model in its various configurations with recent studies on phishing detection. This comparison highlights the effectiveness and advancements achieved by the proposed model.

Scenario #1 :

In this scenario, a step-by-step comparison was carried out to measure the performance of various classifiers individually. These results were then benchmarked against the K-Gen model, which incorporates clustering and genetic-based feature

selection. Subsequently, the final model, namely K-Gen PhishGuard, was compared with all earlier versions to assess its overall effectiveness.

• Case 1: Comparison between manual classifiers and classifier with GA

The performances of classifiers, as demonstrated in Table 1, are shown to be significantly enhanced when Genetic Algorithm (GA) is utilised for feature selection. A marked improvement in accuracy, precision and F1-score is observed across most classifiers. By automating the selection of the most important features, dimensionality is reduced and the learning process simplified. Without the application of GA, classifiers process the entire dataset, which results in lower performance metrics. However, with GA, more effective feature combinations are obtained, leading to improved classification outcomes without overfitting.

For the SVM classifier, accuracy increased from 0.9466 to 0.9500, with precision and F1-score improving from 0.9492 to 0.9514, and 0.9458 to 0.9512, respectively. A remarkable enhancement is observed for the Random Forest classifier, where accuracy rises from 0.9466 to 0.9755, and precision and F1-score improve from 0.9492 to 0.9755 and 0.9458 to 0.9754, respectively. This improvement is attributed to GA's ability to optimise ensemble learning. Similarly, AdaBoost shows improved performance, with accuracy increasing from 0.9536 to 0.9597, and precision and F1-score rising from 0.9531 to 0.9598 and 0.9531 to 0.9597, respectively. XGBoost also demonstrates enhanced metrics, with accuracy improving from 0.9733 to 0.9770, and precision and F1-score increasing from 0.9742 to 0.9769 and 0.9729 to 0.9769, respectively. These results underscore the critical role of GA in optimising feature selection, significantly enhancing the accuracy, precision and F1-scores of various classifiers, rendering it an effective method for improving classification performance—particularly for large datasets with many features.

From Fig. 2, applying the classifiers to high-quality and balanced datasets can be concluded to yield acceptable values of accuracy, precision and F1-score. However, in Fig. 3, all metrics are observed to have increased, especially for the SVM and Random Forest classifiers. A substantial improvement is gained by the SVM classifier, whilst even higher accuracy enhancement is achieved by the Random Forest classifier through the application of an ensemble learning technique. In the latter, multiple decision trees are constructed to improve decision-making performance.

Table 1,
Comparisons between classifiers before and after using (GA)

Without Using Genetic Algorithm				Using Genetic Algorithm for Feature Selection			
Classifier	**Acc**	**P**	**F1**	**Classifier**	**Acc**	**P**	**F1**
SVM	94.6%	94.9%	94.5%	SVM	95.1%	95.1%	95.1%
Random Forest	94.6%	94.9%	94.5%	Random Forest	97.5%	97.5%	97.5%
AdaBoost	95.3%	95.3%	95.3%	AdaBoost	96.0%	96.0%	96.0%
XGBoost	97.3%	97.4%	97.2%	XGBoost	97.7%	97.7%	97.7%

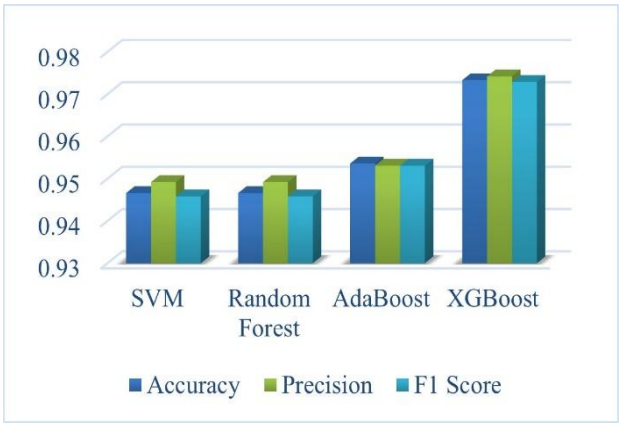

Fig. 3. Classifier results without using GA

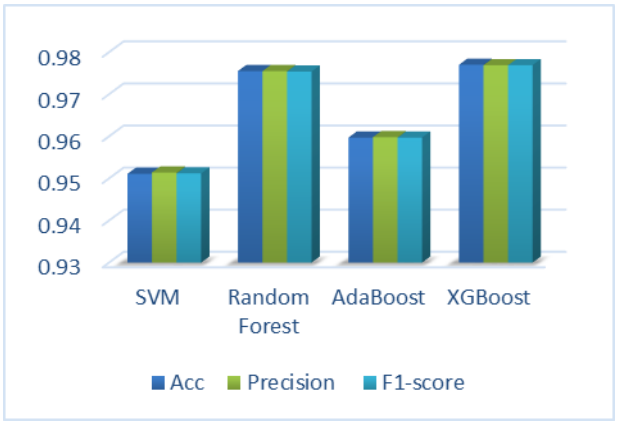

Fig. 4. Classifier results using GA

• Case 2: Comparison between classifier clusters using the K-Gen model

To further optimise and enhance the phishing detection model, additional steps were introduced to ensure greater robustness and accuracy. One key advancement in the approach was the incorporation of K-means clustering into the feature selection process. K-means, an unsupervised learning algorithm, played a vital role in classifying and grouping similar data points based on their underlying characteristics. By leveraging K-means, the complexity of the dataset was reduced by clustering similar URL features. This step not only helps in minimising redundant information but also assists in identifying meaningful patterns that differentiate phishing websites from legitimate ones. Figs. (5–8) illustrate the clustering process and its impact on feature selection and classification. Accordingly, it ensures that the subsequent classification models operate more efficiently, focusing on the most relevant features, leading to enhanced performance in phishing detection. The use of K-means is significantly contributed to building a more accurate and robust model, as shown in Table 2.

The high performance of the proposed model, as demonstrated by the results presented in Table 4, is attributed to the effective use of K-means clustering. In this model, the optimal number of clusters (K) was determined to be 4, which was identified using the Within-Cluster Sum of Squares (WCSS) method. WCSS was used to evaluate the compactness of clusters by minimising intra-cluster variance, ensuring that similar data points were grouped closely together. To further confirm this optimal choice of K, the elbow method was applied, and the WCSS was plotted against varying K values. The ‘elbow’ point, where the curve begins to flatten, was used to indicate the most appropriate number of clusters, which in this case was found to be 4. This separation of clusters is shown to enhance the model’s ability to isolate distinct patterns in phishing and legitimate websites, thereby boosting classification accuracy and leading to the strong metrics that were achieved.

XGBoost is seen to perform the best overall and tops the accuracy (97.69), precision (97.6) and F1 score (97.69) amongst the classifiers, especially in Cluster 1. Such exceptional performance is due to the fact that the clustering process is capable of effectively grouping its data points, thus allowing XGBoost to optimise its decision-making process.

The high performance of the proposed model, as demonstrated by the results presented in Table 4, is attributed to the effective use of K-means clustering. In this model, the optimal number of clusters (K) was determined to be 4, which was identified using the Within-Cluster Sum of Squares (WCSS) method. WCSS was used to evaluate the compactness of clusters by minimising intra-cluster variance, ensuring that similar data points were grouped closely together. To further confirm this optimal choice of K, the elbow method was applied, and the WCSS was plotted against varying K values. The ‘elbow’ point, where the curve begins to flatten, was used to indicate the most appropriate number of clusters, which in this case was found to be 4.

Table 2,
Comparative Analysis for Each Classifier with K-means clustering

Classifiers	Cluster-No	Acc	P	F1
Random forest	1	97.5%	97.5%	97.5%
	2	95.9%	95.9%	95.9%
	3	95.9%	95.9%	95.8%
	4	95.5%	95.5%	95.4%
SVM	1	95.1%	95.1%	95.1%
	2	93.2%	93.3%	93.1%
	3	94.0%	94.0%	94.0%
	4	93.4%	93.3%	93.4%
Adboost	1	95.9%	95.9%	95.9%
	2	95.1%	95.2%	95.1%
	3	96.0%	96.0%	96.0%
	4	94.8%	94.7%	94.7%
XGBoost	1	97.6%	97.6%	97.6%
	2	95.9%	95.9%	95.9%
	3	96.0%	96.0%	96.0%
	4	95.5%	95.4%	95.4%

This separation of clusters is shown to enhance the model's ability to isolate distinct patterns in phishing and legitimate websites, thereby boosting classification accuracy and leading to the strong metrics that were achieved.

XGBoost is seen to perform the best overall and tops the accuracy (97.6), precision (97.69) and F1 score (97.6) amongst the classifiers, especially in Cluster 1. Such exceptional performance is due to the fact that the clustering process is capable of effectively grouping its data points, thus allowing XGBoost to optimise its decision-making process.

Similarly, the Random Forest classifier produces strong results. In Cluster 1, the Accuracy is 97.5 with the Precision and F1 score at 97.55 and 97.5, respectively. In addition, Random Forest, as an ensemble learning type of learning, together with the feature selection achieved by K-means, is responsible for this robust performance. However, its performance somewhat decreases in the following clusters, with the lowest metrics of Cluster 4.

Notably, the K-Gen model, which combines the K-means clustering approach with feature selection, has played a critical role in enhancing the classification results for all classifiers. By segmenting the data into meaningful clusters and selecting the most relevant features, the classifiers successfully focused on the most impactful attributes, resulting in improved performance metrics.

• Case 3: Comparison between voting with GA and voting without the K-Gen model

In the final case, the phishing detection model was enhanced by incorporating a voting method within an ensemble learning framework, combined with a genetic algorithm, to create a robust detection system. The results were compared to the same model without the use of the genetic algorithm, as presented in Table 3.

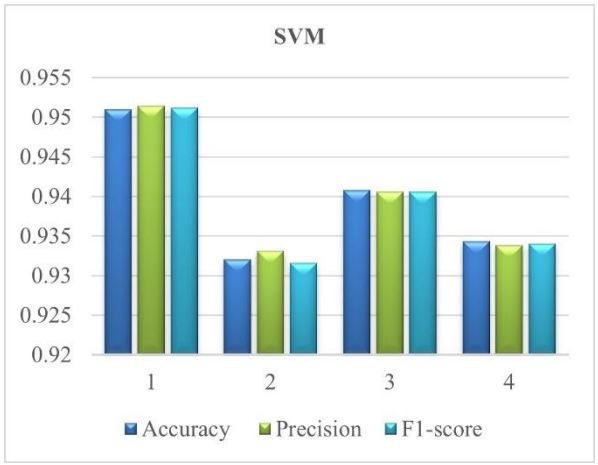

Fig. 5. SVM evaluation with clustering

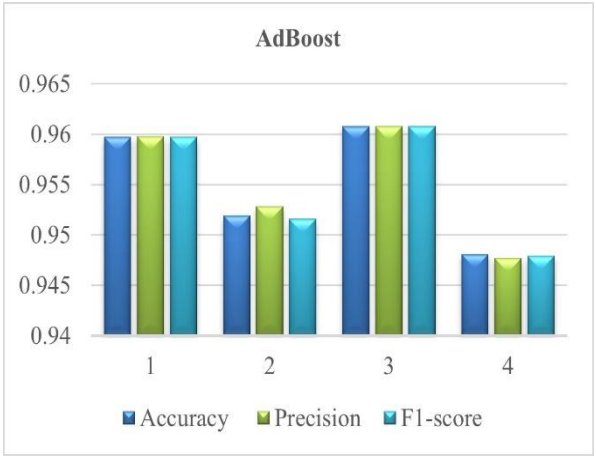

Fig. 6. AdaBoost evaluation with clustering

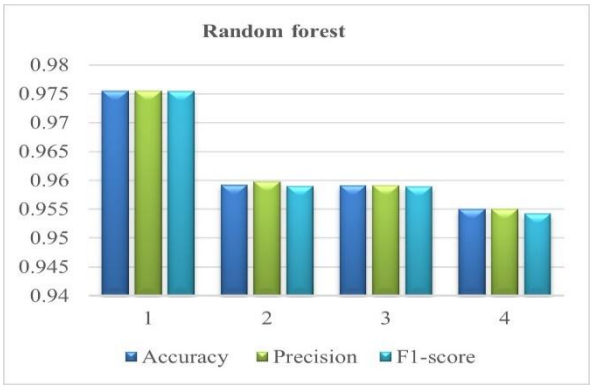

Fig. 7. Random Forest evaluation with clustering

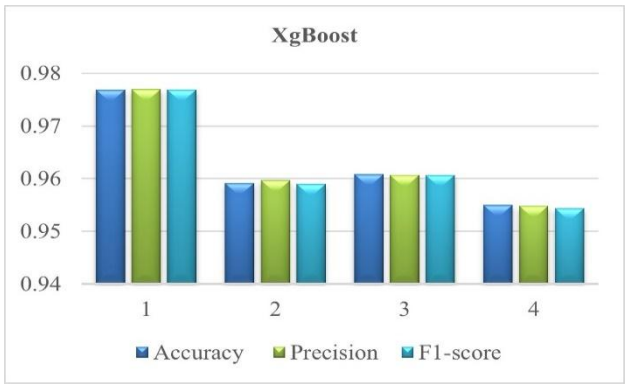

Fig. 8. XGBoost evaluation with clustering

The comparison presented in Table 3 may be initially perceived as unfair by researchers, as the first model was applied to aggregated data based on similarity and relevant features, whereas the second model was applied to unaggregated data without considering similarity. Therefore, this comparison primarily aims to evaluate the same model on the same dataset but without utilising methods for selecting influential features or choosing an appropriate classifier. The results are assessed, as illustrated in Figs. 9 and 10.

A highly accurate phishing detection model that avoids bias towards false positives and effectively distinguishes between phishing and legitimate websites was achieved by implementing a key development step by utilising voting ensemble learning to strategically combine the strengths of diverse classifiers, such as Random Forest, SVM, XGBoost and AdBoost, thereby enhancing overall accuracy and resilience by aggregating predictions and mitigating the biases and limitations of individual algorithms. Therefore, a genetic algorithm is used again but this time to select the classifier, depending on the metrics result of that classifier.

A key point that many researchers ask, is the ratio used to split the dataset.

The first model used k = 5-fold cross validation, where the training set was partitioned into 5 subsets and used as follows: one subset as test set and the other subsets as train set. By using this method, the model is evaluated over the different data splits which help in adding robustness to the model (resistant to computational artifacts) and less biased in terms of performance evaluation. By contrast, the second model employed a 70:I split 30, where 70% of the data are used for training and 30% for testing. This simple split clearly separates the training from the evaluation phases whilst allowing performance of the model to be tested on data that are entirely unseen.

The large difference in performance is shown in the table when the Voting model with K-Gen is used versus without K-Gen. After K-Gen is applied, high metrics are achieved for all clusters, with accuracy, precision and F1 score constantly being above 98%. All metrics in Cluster 1 are attained at 99% performance. In comparison, the Voting model without K-Gen results in an accuracy of 77.3%, precision of 82.1% and F1 score of 76%.

Table 3,
Comparison of the voting model with and without K-Gen

Model	Cluster-No	Accuracy	Precision	F1-Score
Voting with K-Gen	1	99%	99%	99%
	2	98.6%	98.6%	98%
	3	98.9%	98.9%	98%
	4	98.5%	98.5%	98%
Voting without K-Gen	N/A	77.3%	82.1%	76%

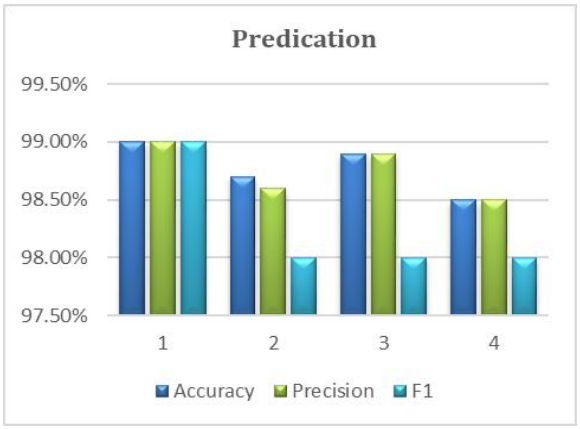

Fig. 9. Voting with K-Gen Model

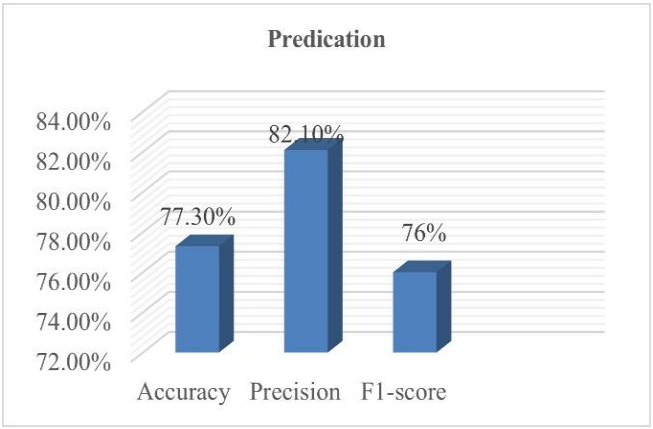

Fig. 10. Voting without K-Gen Model

The variation in these results is attributed primarily to the fact that the Genetic Algorithm (GA) was applied twice in the proposed model, a practice that has not been commonly observed in most studies addressing phishing detection. In the first instance, GA was utilised to search for the optimal or near-optimal solution by identifying the most impactful features that assist the classifier, even if its classification performance was initially suboptimal. In the second instance, GA was employed to select the most effective classifier for the classification process during the ensemble classification phase. This dual application of GA has contributed significantly to the superior results achieved in the proposed model.

Scenario #2:

In this scenario, the performance of the proposed model is compared to studies that utilise the same dataset. The performance of the model is structured into different cases, wherein a particular attribute of the model's performance is considered in each case and its benefits explained. The first case focuses on assessing the effectiveness of the proposed model under identical dataset conditions, emphasising key performance metrics.

• Case 1:

To justify the performance of the proposed phishing detection method, it is compared with studies that utilise the same dataset. [47] The study focuses on establishing reproducible benchmarks for phishing detection using a dataset of 11,430 samples with 87 features spanning URL characteristics, web page content and external service data. The authors employed feature selection methods, namely filter and wrapper methods, for identifying significant improvements in model performance when using filter-based incremental feature ranking. Amongst the classifiers evaluated, Random Forest achieved the highest accuracy of 96.83%, surpassing other models such as SVM, KNN and Logistic Regression. The study underscores the importance of feature selection in enhancing detection accuracy whilst reducing computational overhead, providing a robust framework for comparing phishing detection methodologies, as shown in Table 4.

Table 4 presents a comparison on the same dataset, with an identical number of samples and features but different feature selection methods applied. The referenced study employed the Chi-square method to select features and improve the classifier's performance in phishing detection. The results were satisfactory, utilising two evaluation metrics, namely accuracy and F1-score. Amongst the tested classifiers, the Random Forest algorithm achieved the highest performance in this experiment.

This comparison primarily aims to demonstrate the efficiency of the Genetic Algorithm in feature selection, as it significantly outperformed the Chi-square method. Notably, the proposed model is in its initial development stage, focusing solely on the Genetic Algorithm for feature selection without incorporating K-Means clustering or ensemble learning techniques.

• Case 2:

This study [15] utilised the same proposed dataset and was divided and processed using the 10-fold cross-validation method. This approach enhanced the model's efficiency and robustness. Subsequently, the Recursive Feature Elimination (RFE) technique was employed to identify the most representative subset of features for the dataset.

RFE iteratively removed the least significant features based on model performance until the optimal feature set was achieved.

Table 4,
Comparison of feature selection methods across classifiers

Ref.	Feature selection Method	Classifiers	Accuracy	F1-score
[47]	Chi-square	DT	94.13%	94.10%
		RF	96.83%	96.60%
		LR	94.48%	94.50%
		NV	79.80%	79.80%
		SVM	73.95%	72.10%
Proposed Model	Genetic Algorithm	RF	97.55%	97.54%
		XGBoost	97.70%	97.69%
		SVM	95.11%	95.12%
		AdaBoost	95.97%	95.97%

Once the feature selection process was completed, the refined data were fed into the classifiers, including Support Vector Machines (SVM), Random Forest (RF), XGBoost (XGB) and AdaBoost (Ada).

The results were then compared using evaluation metrics such as accuracy, F1-score and precision to determine the performance of the proposed framework, as shown in Table 5.

In the table above, the comparison primarily relies on feature selection methods, despite the similarity in the datasets and classifiers used. However, the highest accuracy, F1-score and precision were achieved by the XGB classifier.

Nonetheless, all classifiers in the proposed phishing detection model notably outperform the results in terms of accuracy, F1-score and precision.

• Case 3:

This study [47] used datasets from different repositories, and a sophisticated stacking ensemble classifier was proposed as the model. The proposed model was compared with its final enhancement, namely K-Gen Phish Guard. Table 6 compares with two different models that utilise the same dataset, (Dataset-4) consisting of 11, 430 samples and 87 features. In the first model, a Sophisticated Framework, a combination of feature selection methods, greedy algorithms, cross-validation and deep learning techniques were used to build a stacking ensemble classifier. The highest performance was achieved on Dataset-4, with an accuracy of 98.20%. In the second proposed model (K-Gen Phish Guard), genetic algorithms were used once again—this time to select the best-performing classifiers in combination with soft voting ensemble learning. This model demonstrated high efficiency and performance, achieving 99% in accuracy, F1-score and precision across most clusters and outperforming the first model in nearly all aspects.

Table 5,
Comparison between RFE and GA across classifiers

Ref	Feature Selection	Classifiers	Acc	P	F1
[15]	RFE	SVM	95.9%	96.2%	95.8%
		RF	96.6%	96.8%	96.6%
		AdaBoost	94.6%	94.9%	94.6%
		XGBoost	97.0%	96.9%	97.0%
Proposed Model	GA	RF	97.5%	97.5%	97.5%
		XGBoost	97.7%	97.6%	97.6%
		SVM	95.1%	95.1%	95.1%
		AdaBoost	95.9%	95.9%	95.9%

Table 6,
Comparison of 'A Sophisticated Framework' and 'K-Gen Phish Guard' Across Datasets and Clusters

Ref	Mode	Dataset/Classifier	Cluster-No	Acc	P	F1
[48]	Single and Hybrid-Ensemble Learning-Based Phishing Website Detection	Dataset-4/Stacking	-	98.20%	94.85%	94.45%
Proposed Model	(K-Gen Phish Guard)	Dataset-4/Voting	1	99%	99%	99%
			2	98.6%	98.6%	98%
			3	98.9%	98.9%	98%
			4	98.5%	98.5%	98%

7. Conclusion

An effective and powerful approach for detecting phishing sites through two scenario comparisons were presented. In the first comparisons, the classifiers were evaluated before and after the genetic algorithm was used. Remarkable progress in the accuracy and efficiency of the model in detecting phishing was observed, which was referred to as the first developmental stage. Subsequently, the second developmental stage involved integrating unsupervised learning, specifically k-means, with the genetic algorithm to divide the dataset into clusters. Within each cluster, a set of influential attributes specific to that cluster was identified to facilitate and enhance the speed and efficiency of the classifier in detecting phishing. This improvement was evident in most clusters, with advancements in accuracy, F1-score and precision compared with classifiers

that only used the genetic algorithm without k-means.This model was named K-GEN. The third developmental stage introduced the proposed model, namely K-Gen Phish Guard, which outperformed all benchmark models. It achieved an accuracy rate of 99%, demonstrating superior performance compared with all previous models. A second comparison was made between the proposed K-Gen Phish Guard model and recent studies on phishing site detection that used different feature selection methods and classifiers but were based on the same dataset. The proposed model consistently outperformed all models mentioned in the earlier studies.

In the model, the dataset was divided into a 7:3 ratio, with 70% allocated for training and 30% for testing, ensuring a robust evaluation process. This approach demonstrated a stronger ability to detect phishing whilst providing a scalable solution to various real-world cybersecurity challenges. In the future, this model could be applied to other types of cyber threats or adapted to counter emerging phishing techniques, further enhancing its role in mitigating cybersecurity risks and addressing additional aspects of phishing defense.

Abbreviations

Acc	Accuracy
AdaBoost	Adaptive boosting
ANN	Artificial neural network
Chi-square	A chi-square statistic
CNN	Convolutional neural network
DL	Deep Learning
DNN	Deep neural networks
DT	Decision Tree
FN	False Negative
FP	False Positive
IG	Information Gain
ML	Machine learning
NB	Nave Bayesian
P	Precision
PCA	Principal Component Analysis
R	Recall
RF	Random forest
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
URL	Uniform resource locator
XGBoost	Extreme Gradient Boosting

References

[1] M. S. Bakken, "Webpage Fingerprinting using Infrastructure-based Features," NTNU, 2023.

[2] P. Patel, D. M. Sarno, J. E. Lewis, M. Shoss, M. B. Neider, and C. J. Bohil, "Perceptual representation of spam and phishing emails," *Applied Cognitive Psychology*, vol. 33, no. 6, pp. 1296-1304, 2019.

[3] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse, "Phishing attacks and defenses," *International journal of security and its applications*, vol. 10, no. 1, pp. 247-256, 2016.

[4] M. A. Chargo, "You've been hacked: How to better incentivize corporations to protect consumers' data," *Transactions: The Tennessee Journal of Business Law*, vol. 20, pp. 115-143, 2018.

[5] G. Ho et al., "Understanding the Efficacy of Phishing Training in Practice," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2024: IEEE Computer Society, pp. 76-76.

[6] R. A. Al Mudhafar and N. K. El Abbadi, "Image Noise Detection and Classification Based on Combination of Deep Wavelet and Machine Learning," *Al-Salam Journal for Engineering and Technology*, vol. 3, no. 1, pp. 23-36, 2024.

[7] L. Al-Shalabi and Y. Hasan Jazyah, "Phishing Detection Using Hybrid Algorithm Based on Clustering and Machine Learning," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1-13, 2024.

[8] G. Sonowal and K. Kuppusamy, "PhiDMA–A phishing detection model with multi-filter approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 99-112, 2020.

[9] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153-166, 2019, doi: 10.1016/j.ins.2019.01.064.

[10] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network," *Wireless Communications and Mobile Computing*, vol. 2021, p. 24, 2021, doi: 10.1155/2021/8241104.

[11] J. Solanki and R. G. Vaishnav, "Website phishing detection using heuristic based approach," in *Proceedings of the third international conference on advances in computing, electronics and electrical technology*, 2015.

[12] L. A. T. Nguyen and H. K. Nguyen, "Developing an efficient fuzzy model for phishing identification," in *2015 10th Asian Control Conference (ASCC)*, 2015: IEEE, pp. 1-6.

[13] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443-458, 2014.

[14] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in *2012 international conference for internet technology and secured transactions*, 2012: IEEE, pp. 492-497.

[15] M. S. I. Ovi, M. H. Rahman, and M. A. Hossain, "PhishGuard: A Multi-Layered Ensemble Model for Optimal Phishing Website Detection," *arXiv preprint arXiv:2409.19825*, 2024.

[16] A. R. Mahmood and S. M. Hameed, "A Smishing Detection Method Based on SMS Contents Analysis and URL Inspection Using Google Engine and VirusTotal," *Iraqi Journal of Science*, pp. 6276-6291, 2023.

[17] A. R. Mahmood and S. M. Hameed, "Review of Smishing Detection Via Machine Learning," *Iraqi Journal of Science*, pp. 4244-4259, 2023.

[18] A. A. Zuraiq and M. Alkasassbeh, "Phishing detection approaches," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 2019: IEEE, pp. 1-6, doi: 10.1109/ICTCS.2019.8923069 .

[19] M. Pratiwi, T. Lorosae, and F. Wibowo, "Phishing site detection analysis using artificial neural network," *Journal of Physics: Conference Series*, vol. 1140, p. 012048, 2018, doi: 10.1088/1742-6596/1140/1/012048.

[20] A. Odeh, I. Keshta, and E. Abdelfattah, "PHIBOOST-a novel phishing detection model using Adaptive boosting approach," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 7, no. 1, pp. 65-74, 2021.

[21] H. Shirazi, K. Haefner, and I. Ray, "Improving auto-detection of phishing websites using fresh-phish framework," *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 9, no. 1, p. 14, 2018, doi: 10.4018/IJMDEM.2018010104.

[22] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, p. 15, 2019, doi: 10.1155/2019/2595794.

[23] Z. Liu, B. Yang, J. An, and C. Huang, "Similarity evaluation of graphic design based on deep visual saliency features," *The Journal of Supercomputing*, pp. 1-22, 2023.

[24] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308-6325, 2020.

[25] S. Juanita and R. D. Cahyono, "K-means clustering with comparison of Elbow and silhouette methods for medicines clustering based on user reviews," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 1, pp. 283-289, 2024.

[26] S. Mathankar, S. R. Sharma, T. Wankhede, M. Sahu, and S. Thakur, "Phishing Website Detection using Machine Learning Techniques," in *2023 11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP)*, 2023: IEEE, pp. 1-6.

[27] R. Mahajan and I. Siddavatam, "Phishing website detection using machine learning algorithms," *International Journal of Computer Applications*, vol. 181, no. 23, pp. 45-47, 2018.

[28] A. Altaher, "Phishing websites classification using hybrid SVM and KNN approach," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017.

[29] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[30] D. M. Abdullah and A. M. Abdulazeez, "Machine learning applications based on SVM classification a review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 81-90, 2021.

[31] A. Roy and S. Chakraborty, "Support vector machine in structural reliability analysis: A review," *Reliability Engineering & System Safety*, vol. 233, p. 109126, 2023.

[32] A. Parmar ,R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, 2019: Springer, pp. 758-763.

[33] W. Wang and D. Sun, "The improved AdaBoost algorithms for imbalanced data classification," *Information Sciences*, vol. 563, pp. 358-374, 2021.

[34] S. S. Azmi and S. Baliga, "An overview of boosting decision tree algorithms utilizing AdaBoost and XGBoost boosting strategies," *Int. Res. J. Eng. Technol*, vol. 7, no. 5, pp. 6867-6870, 2020.

[35] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 8, no. 4, p. e1249, 2018.

[36] J. Tang, S. Alelyani, and H. Liu" ,Feature selection for classification: A review," *Data classification: Algorithms and applications*, p. 37, 2014.

[37] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and information technologies*, vol. 19, no. 1 ,pp. 3-26, 2019.

[38] S. N. Mohammed and A. J. Jabir, "A Ranked-Aware GA with HoG Features for Infant Cry Classification," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 6, 2023.

[39] A. Sohail, "Genetic algorithms in the fields of artificial intelligence and data sciences," *Annals of Data Science*, vol. 10, no. 4, pp. 1007-1018, 2023.

[40] W. Ali and F. Saeed, "Hybrid filter and genetic algorithm-based feature selection for improving cancer classification in high-dimensional microarray data," *Processes*, vol. 11, no. 2, p. 562, 2023.

[41] X. Liu and Y. Du, "Towards effective feature selection for iot botnet attack detection using a genetic algorithm," *Electronics*, vol. 12, no. 5, p. 1260, 2023.

[42] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091-8126, 2021.

[43] G. K. Soon, T. T. Guan, C. K. On, R. Alfred, and P. Anthony, "A comparison on the performance of crossover techniques in video game," in *2013 IEEE international conference on control system, computing and engineering*, 2013: IEEE, pp. 493-498.

[44] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9 ,no. 1, pp. 381-386, 2020.

[45] G. Ascenso, M. H. Yap, T. Allen, S. S. Choppin, and C. Payton, "A review of silhouette extraction algorithms for use within visual hull pipelines," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 8, no. 6, pp. 649-670, 2020.

[46] Kaggle, "Web page Phishing Detection Dataset," 2021. [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset

[47] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104347, 2021.

[48] K. Adane, B. Beyene, and M. Abebe, "Single and hybrid-ensemble learning-based phishing website detection: examining impacts of varied nature datasets and informative feature selection technique," *Digital Threats: Research and Practice*, vol. 4, no. 3, pp. 1-27, 2023.

K-Gen PhishGuard
نهج متكامل للكشف عن التصيد الاحتيالي باستخدام K-Means والخوارزمية الجينية

علي رحيم الحافظ [١] *، عدنان جمعه جابر [٢] ، شمالة سوبرامانيام [٣]

[٢،١] قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد ، العراق
[٣] قسم تكنولوجيا الاتصالات والشبكات، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة بوترا ماليزيا، سيردانج، ٤٣٤٠٠، سيلانجور دار الإحسان، ماليزيا
*البريد الالكتروني : aliraheem2201m@sc.ubaghdad.edu.iq

المستخلص

الكشف عن التصيد الاحتيالي هو مشكلة حرجة في مجال الأمن السيبراني، وأكبر تحدٍ هو كيفية استخدام التعلم الآلي مع طريقة فعالة لاختيار الميزات لتحديد المواقع الضارة بدقة. يقدم هذا البحث نظامًا للكشف عن التصيد الاحتيالي يتكون من مرحلتين رئيسيتين، يتم فيهما استخدام اختيار الميزات غير المراقب والتصنيف المراقب.في المرحلة الأولى، تُستخدم خوارزمية التحسين الجيني(GA) لتحديد أفضل مجموعة من الميزات التي يتم استخدامها بواسطة خوارزمية التجميع K-means لتقسيم مجموعة البيانات إلى مجموعات تحمل سمات متشابهة.أما في المرحلة الثانية، فيتم استخدام خوارزمية التحسين الجيني (GA) مرة أخرى لتحديد أفضل مجموعة ميزات داخل كل مجموعة، مما يعزز عملية التصنيف. في النهاية، يتم تطبيق تقنية التجميع بالتصويت Voting (Ensemble، حيث يتم دمج نماذج Support Vector Machine (SVM) وRandom Forest (RF) وXGBoost وAdaBoost) باستخدام آلية تصويت ناعمة لتجميع التنبؤات.تم استخدام مجموعة بيانات خاصة بالكشف عن التصيد الاحتيالي لصفحات الويب في هذا البحث، تحتوي على ١١,٤٣٠ عنوان URL و٨٧ ميزة.أظهرت النتائج أن تقنية التجميع بالتصويت تحقق دقة تصل إلى ٩٩٪ عند استخدام اختيار الميزات، مقارنة بـ ٧٧,٣٪ دون استخدام اختيار الميزات. يُظهر اختيار الميزات المُحسَّن باستخدام خوارزمية (GA) تحسينًا كبيرًا في أداء النموذج، من خلال تقليل التعقيد الحسابي وتحسين المؤشرات الرئيسية مثل الدقة ((Accuracy، ومعامل التحديد ((Precision، ودرجة .F1 (F1-score) إضافة إلى ذلك، تُظهر النتائج عبر أربع مجموعات البيانات أن خوارزمية K-means تُسهم بشكل إيجابي في تحسين دقة التصنيف ضمن مجموعات بيانات معينة. تُثبت النتائج المحققة أن دمج اختيار الميزات مع تقنيات التعلم المجمع يعد حلاً فعالاً للكشف عن التصيد الاحتيالي، ويظهر قابلية التطبيق والكفاءة لهذا الحل في الاستخدامات الواقعية.