


Phishing detection using clustering and machine learning

Luai Al-Shalabi, Yahia Hasan Jazyah
Faculty of Computer Studies, Arab Open University, Ardiya, Kuwait

| Article Info | ABSTRACT |
|--|---|
| <p>Article history:</p> <p>Received Dec 28, 2023 Revised May 16, 2024 Accepted Jun 1, 2024</p> | <p>Phishing is a prevalent and evolving cyber threat that continues to exploit human vulnerability to deceive individuals and organizations into revealing sensitive information. Phishing attacks encompass a range of tactics, from deceptive emails and fraudulent websites to social engineering techniques. Traditional methods of detection, such as signature-based approaches and rule-based filtering, have proven to be limited in their effectiveness, as attackers frequently adapt and create new, previously unseen phishing campaigns. Consequently, there is a growing need for more sophisticated and adaptable detection methods. In recent years, machine learning (ML) and artificial intelligence (AI) have played a significant role in enhancing phishing detection. These technologies leverage large datasets to train models capable of recognizing subtle patterns and anomalies in both email content and website behavior. This research proposes a hybrid algorithm to detect phishing attacks based on clustering and classification machine learning methods (CMLM): deep learning (DL) and decision tree (DT). Simulation results show that the proposed technique achieves a high percentage of accuracy in detecting phishing.</p> |
| <p>Keywords:</p> <p>Artificial intelligence Decision tree Deep learning Machine learning Phishing</p> | |
| | <p><i>This is an open access article under the CC BY-SA license.</i></p> |
| |  |
| <p>Corresponding Author:</p> <p>Yahia Hasan Jazyah Faculty of Computer Studies, Arab Open University St. Mohammed Nazzal Al-Moassab, Ardiya, Kuwait Email: yahia@aou.edu.kw</p> | |

1. INTRODUCTION

Phishing is a pervasive and insidious form of cybercrime that preys on human psychology and technical vulnerabilities. It involves the use of deceptive techniques to trick individuals or organizations into divulging sensitive information, such as login credentials, financial details, or personal data. Phishing attacks are often the initial entry point for broader cyber threats, including identity theft, fraud, and malware infections. To combat this growing menace, effective phishing detection methods have become indispensable.

The sophistication of phishing attacks continues to evolve, making it a challenging task to thwart these threats. Cybercriminals utilize a variety of tactics, including misleading emails, fraudulent websites, and social engineering strategies that exploit human trust and curiosity. The dynamic nature of these attacks means that traditional, static security measures are often ineffective. This has led to the development of advanced and adaptive techniques for detecting and mitigating phishing attempts.

Phishing detection involves the identification and prevention of deceptive or malicious content within emails, websites, or other digital communication channels. It encompasses a broad spectrum of methods, ranging from rule-based filters and signature-based systems to more advanced approaches that leverage artificial intelligence (AI), machine learning (ML), and behavioral analysis. As cybercriminals constantly refine their tactics to bypass conventional defenses, the need for innovative and responsive detection mechanisms has become increasingly pressing.

In this context, this paper explores the landscape of phishing detection, addressing both the existing challenges and the latest advancements in the field. And proposing a hybrid algorithm that merges between clustering and classification using deep learning (DL) and decision tree (DT) for two different datasets (DSs).

The main contributions of this work are summarized in three-fold:

- A robust hybrid algorithm to detect phishing attacks using clustering, classification, and stability-correlation and correlation (ScC) feature selection methods was proposed considering the speed and simplicity.
- Thorough statistical analysis of the proposed method using well-known phishing datasets.
- A comparison between the proposed method and other well-known methods for detecting phishing websites that are presented in the literature, methods that use DL and DT together with ScC, rough set (RS), and principal component analysis (PCA), and methods that use DL and DT together with RS-K-mean and PCA-K-mean.

The remaining of this article is organized as follows: section 2 presents comparisons between phishing detection Algorithms, section 3 is preliminaries about feature selection methods, section 4 presents the proposed algorithm, section 5 presents the complexity of the proposed algorithm, and section 6 is the conclusion.

2. PHISHING DETECTION ALGORITHMS' COMPARISONS

In literature, plenty of phishing detection algorithms were developed. It is important to understand the advantages and disadvantages of them. Table 1 summarises these advantages and disadvantages, while Table 2 presents a comparison between different methods for phishing website detection methods in terms of accuracy, which is a metric that measures how well a phishing detection system or algorithm correctly identifies and classifies phishing emails or websites.

3. PRELIMINARIES

3.1. Feature selection methods

Feature selection methods for reducing the size of datasets are classified into three groups: filter, wrapper, and embedded [1], [2]. The filter method calculates a score for each feature and all features with scores more than a pre-defined threshold value are chosen. On the other hand, wrapper methods use a classifier to evaluate the effectiveness of various reducts and choose the best of them. It is more powerful than filter methods, but it is also more complex. Conversely, to wrapper methods, embedded methods judge feature selection in the training procedure. Filter methods were applied for feature selection to select the most presenting attributes that have the highest information in a dataset that can distinguish between classes. The applied methods are explained next. Various simple ML algorithms may work jointly (known as hybrid) to complement and enhance each other, Table 3 presents a comparison between different hybrid methods for phishing website detection methods in terms of accuracy.

Table 1. Pros and cons of phishing detection algorithms

| Algorithm | Advantage | Disadvantage |
|--|--|--|
| Rule-based detection | Simple to implement, can be effective for known phishing patterns | Limited to predefined rules, struggles with new and evolving phishing tactics |
| Signature-based detection | Effective for known phishing threats, can quickly identify known patterns | Ineffective against zero-day attacks, cannot adapt to new tactics |
| ML and AI-based detection | Effective at detecting evolving and new phishing threats, can adapt to changing tactics, can analyze large datasets for patterns | Requires substantial data for training, may be vulnerable to adversarial attacks |
| Behavioural analysis | Effective at identifying anomalous behaviour, can detect zero-day attacks | May produce false positives, can be complex to implement |
| URL analysis | Can detect deceptive URLs and domain spoofing | Limited to URL analysis, may not detect other aspects of phishing. |
| Content analysis | Effective at detecting deceptive language and tactics in emails and websites | May not detect more sophisticated phishing attacks |
| Blacklists and reputation-based approaches | Quick to implement, can block known malicious entities | Ineffective against new threats, may produce false positives |
| Heuristic-based detection | Effective at identifying suspicious forms and requests for sensitive information | May produce false positives, limited to heuristic-based rules |
| Real-time analysis | Can detect and block phishing attacks in real time | May require substantial computational resources, can be resource-intensive |
| Hybrid approaches | Combine multiple detection methods to improve accuracy | May be more complex to implement, require ongoing tuning |

Table 2. Accuracy of non-traditional methods for phishing websites detection

| Anti-phishing method | Authors | Techniques | Dataset | Accuracy (%) |
|----------------------|-----------------------------|--|--|--------------|
| Content-based | Jain <i>et al.</i> [3] | Modified term frequency-inverse document frequency (TF-IDF) | Alexa dataset [4], OpenPhish [5], and Phish Tank [6] | 89 |
| | Sonowal and Kuppusamy [7] | Phishing detection using multi-filter approach (PhiDMA) framework incorporates five layers | Phishload and Legitimate URL dataset [8] | 92.72 |
| Heuristics-based | Rao <i>et al.</i> [9] | Twin support vector machines (TWSVM) | Phish Tank [6] and Alexa dataset [4] | 98.05 |
| | Babagoli <i>et al.</i> [10] | meta-heuristics (HS, SVM) | UCI phishing datasets [11], [12] | 92.80 |
| ML | Chiew <i>et al.</i> [13] | Cumulative distribution function gradient (CDF-g), random forest, SVM, naive bayes, C4.5, JRip, and PART | UCI phishing datasets [11], [12] | 94.6 |
| DL | Yadollahi [14] | XCS | Real URLs | 98.39 |
| | Smadi <i>et al.</i> [15] | Reinforcement learning, neural network | Phishing Corpus [16], Spam Assassin [17], and Phish Tank [6] | 97 |
| | Wei <i>et al.</i> [18] | Convolutional neural networks | Phish Tank [6] and Common Crawl Foundation [19] | 99.98 |
| Data Mining | Smadi <i>et al.</i> [20] | J48 algorithm and C4.5 algorithm | Phishing Corpus [16] and Spam Assassin [17] | 98.87 |
| | Subasi [21] | Random forest | UCI [12] and WEKA [22] | 97.36 |

Table 3. Accuracy of hybrid methods for phishing websites detection

| Anti-phishing method | Authors | Techniques | Dataset | Accuracy (%) |
|----------------------|----------------------------|---|---|--------------|
| Hybrid Methods | Ali and Ahmed [23] | Deep neural networks (DNNs) and genetic algorithm (GA) | UCI phishing websites [12] | 91.13 |
| | Zhu <i>et al.</i> [24] | DT and optimal features based artificial neural network (ANN), K-medoids clustering algorithm | UCI [11], [12], Phish Tank [6], Alexa [4] | 95.76 |
| | Suleman and Awan [25] | Iterative dichotomiser-3 (ID3) and yet another generating genetic algorithm (YAGGA) | UCI ML website [11], [12] | 95 |
| | Vrbanić <i>et al.</i> [26] | Bat algorithm (BA) and hybrid bat algorithm (HBA) | UCI [12] | 96.5 |
| | Chin <i>et al.</i> [27] | Deep packet inspection (DPI), software-defined networking (SDN) and ANN | UCI [12] | 98.39 |
| | Chen <i>et al.</i> [28] | Particle swarm optimization (PSO) and back propagation (BP) neural network | Phishtank [6] | 98.95 |

3.1.1. Rough set theory [29]

It is a mathematical framework and set of principles used for data analysis and feature selection. The core idea of RS is to handle uncertainty and vagueness in data. It works with incomplete, imprecise, or inconsistent information to approximate and reason about data. It identifies the most relevant features in a DS while minimizing information loss. The primary concept behind RS feature selection is to partition the data into equivalence classes based on the values of a particular feature and analyze the dependency of the class labels on that feature. The primary (1) involved in the rough set feature selection method is the dependency score, which measures the importance of a feature in classifying data.

$$Dependency(SI, A) = \frac{|SI|}{|U|} * \frac{|SI| - |SI|_A}{|SI|} \quad (1)$$

Where Dependency (SI, A) is the dependency score of feature A concerning the set of instances SI, SI is the set of instances for which the feature is evaluated, A is the feature for which the dependency is measured, |SI| is the number of instances in SI, |U| is the total number of instances in the dataset, |SI|_A is the number of distinct values of feature A in SI. The dependency score measures the significance of feature A in discriminating between different classes or values of the target variable within the set of instances SI. A higher Dependency Score indicates a stronger dependency, and therefore, the feature is considered more important for classification.

3.1.2. Stability-correlation and correlation [30]

It is a feature selection method used in ML and data analysis. It is designed to identify relevant features by considering both stability (S) which represents the consistency of feature importance based on the variety of the feature's values (high variety of values represents high stability of the feature), and correlation (r) which measures how closely related a feature is to the target variable or not. ScC has two distinct methods for selecting

relevant features, the first method is stability-correlation (Sc) feature selection which combines both stability-based and correlation-based criteria to select features, it aims to identify features that are stable across different subsets of the data and highly correlated with the target variable or class labels. The stability-correlation score is calculated using (2).

$$S = \frac{\text{mode}(Xi)}{n} \quad (2)$$

Where: xi is the feature, n is the number of rows in the dataset.

The second method is correlation-based (CB) feature selection which is applied to the dataset generated by the first method. CB focuses on selecting features that are highly correlated with the target variable while potentially avoiding multicollinearity among the selected features. The equation for assessing the correlation between a feature X and the target variable Y is the Pearson correlation coefficient (PCC) as shown in (3) to (5):

$$r = \frac{1}{n-1} * \sum x \sum y \frac{(x-\bar{x})(y-\bar{y})}{Stx Sty} \quad (3)$$

$$Stx = \sqrt{\sum \frac{(x-\bar{x})^2}{n-1}} \quad (4)$$

$$Sty = \sqrt{\sum \frac{(y-\bar{y})^2}{n-1}} \quad (5)$$

where: n is the number of pairs of data used, Σ is sigma which represents the summation, \bar{x} = the mean of all x-values, \bar{y} is the mean of all y-values, Stx is the standard deviation of variable x, Sty is the standard deviation of variable y.

3.1.3. Principal component analysis [35]

It is a method for reducing the dimensionality of data while preserving as much of the variance in the data as possible. It accomplishes this by transforming the original features (variables) into a new set of linearly uncorrelated variables (principal components). The related concept to PCA is the explained variance (EV), which is used to identify the importance of each principal component. The amount of variance explained by each principal component is a measure of feature importance in a PCA-based feature selection context. In (6) presents the EV for a principal component k.

$$EV(PC_k) = \frac{\text{Eigenvalue}_k}{\text{Total Eigenvalues}} \quad (6)$$

where: EV(PC_k) is the proportion of the total variance explained by the kth principal component (PC), eigenvalue_k is the eigenvalue associated with the kth principal component, and total eigenvalues is the sum of all eigenvalues obtained from PCA.

3.2. Classification methods

ML and AI techniques, such as supervised and unsupervised learning, are employed to build models that can identify phishing attempts based on historical data and patterns. Classification is a supervised ML process of grouping a given dataset into classes based on one or more features. Some common ML algorithms include [9], [11].

3.2.1 Decision trees

It is a supervised ML algorithm used for both classification and regression tasks. It is a popular method for making decisions and solving problems by visually representing a decision-making process as a tree-like structure. Each node in the tree represents a decision or a test on a particular attribute, and each branch represents the outcome of that test. The leaves of the tree contain the final decision or the predicted value.

DTs have several advantages, such as simplicity, interpretability, and ease of visualization. They can handle both categorical and numerical data, and they are capable of handling missing values. However, they can be prone to overfitting, and the structure of the tree may not always generalize well to new data. To mitigate these issues, techniques like pruning and using ensemble methods, such as random forests, are often employed.

3.2.1. Deep learning

It represents a subset of ANNs that consist of multiple layers of interconnected neurons or nodes. These networks are capable of learning complex patterns and representations from large and high-dimensional

datasets, making them a powerful tool for various ML and AI tasks. DL, like convolutional and recurrent neural networks, can analyse email content and patterns in network traffic to detect phishing. However, they can be computationally intensive and require substantial amounts of labeled data for training. Proper architecture selection, hyperparameter tuning, and data preprocessing are crucial for the successful deployment of DL models.

3.3. K-mean clustering method

It is a clustering algorithm, and an unsupervised learning technique designed to partition a dataset into K distinct, non-overlapping clusters. These clusters are characterized by their centroid, which is the mean of the data points within each cluster. K-means divides the dataset into clusters without any hierarchical structure. In our work, we use k=2 since the two datasets are used to distinguish between phishing and non-phishing cases. The following are some reasons why clustering performs better in phishing detection.

Clustering is a technique used for discovering inherent patterns and grouped structures of data. If your data naturally exhibits clusters or groups, the DS in this testing issue can be grouped based on certain criteria. Clustering is an unsupervised learning technique, meaning it does not require prior knowledge or labeled data. It can be used for anomaly detection by identifying data points that don't belong to any of the established clusters. All the above are valid characteristics of DSs belonging to phishing detection.

3.4. Performance measurements

Performance measurements are success indicators that express how well ML algorithms are functioning. To evaluate the performance of the proposed algorithm, a range of measurements can be used. In the work, five of them will be used and explained.

- Accuracy: it is calculated as a ratio represented in (7).

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \quad (7)$$

Where: true positives (TP) is the number of phishing emails or websites correctly identified as phishing, true negatives (TN) is the number of legitimate (non-phishing) emails or websites correctly identified as non-phishing, false positives (FP) is the number of legitimate emails or websites incorrectly identified as phishing (a type of error), false negatives (FN) is the number of phishing emails or websites incorrectly identified as legitimate (another type of error).

- Area under curve (AUC): it is one of the well-known measurements in ML area. It is used to assess the performance of binary classification models. It quantifies the ability of a model to distinguish between two classes (positive and negative) by measuring the area under the receiver operating characteristic (ROC) curve.
- Precision: it is a measure of the accuracy of a model in correctly identifying positive instances among the instances that it has classified as positive TP. It provides information about the ability of model to avoid FP. Precision is calculated by (8).

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

- Recall: it is another vital metric that evaluates ML algorithms. It is known as sensitivity or true positive rate, which is used to assess the performance of a binary classification model by measuring the ability of model to correctly identify all relevant instances from the positive class. It is calculated using (9).

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

- F-measure: it is an important measure to evaluate ML algorithms. It provides a single measure of a classification performance of model by combining both precision and recall into a single score. It is calculated using (10).

$$F - measure = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (10)$$

4. PROPOSED ALGORITHM AND SIMULATION RESULTS

4.1. Datasets

Two phishing datasets [5], [32] are used for training and testing. They are cleaned by removing redundancy and missing values. The number of rows in each DS1 and DS2 are 11,050 and 10,000 respectively.

4.2. Proposed algorithm

The proposed algorithm is a hybrid method that merges feature selection methods, clustering [33], [34] (K-mean), and classification ML methods using the DL (H2O) [35] algorithm, which is an open-source ML platform that is designed for scalable and distributed data analysis. It can perform a wide range of ML tasks efficiently and effectively, particularly for large datasets.

The steps of the process are described in the following:

Step 1. Preparing the datasets

Step 2. Applying feature selection methods (RS, ScC, and PCA) to the original dataset and produce the reduced datasets. Then generate the classification model of each using DL and DT.

Step 3. Testing the performance of the models generated in step 1 using all the metrics explained earlier.

Step 4. Remove the classification attribute from each reduced datasets in step 2.

Step 5. Apply the K-mean to each new dataset generated in step 4.

Step 6. Apply machine learning classification methods (ML and DT) to the resulted datasets in step 5 and generate the models.

Step 7. Testing the performance of the generated hybrid models in step 6 using all the measurements explained earlier.

Step 8. Compare all the results of all the models.

Step 9. Choose the best of them.

The idea was proposed to improve the detection process of phishing data. Figure 1 represents the flow of proposed idea, whereas Figure 2 represents the components of the proposed algorithm. The steps of the process are summarized as in Algorithm 1.

Algorithm 1. Proposed methodology

For each DS_i

Input DS_i

Clean DS_i

For each FS_i

Apply FS_i

Apply ML methods

Output results

//results after feature selection using DL and DT

Apply K-Mean

Apply ML methods

Output results

//results after clustering

//loop until applying all FS_i

End for

//loop until input all DS_i

End for

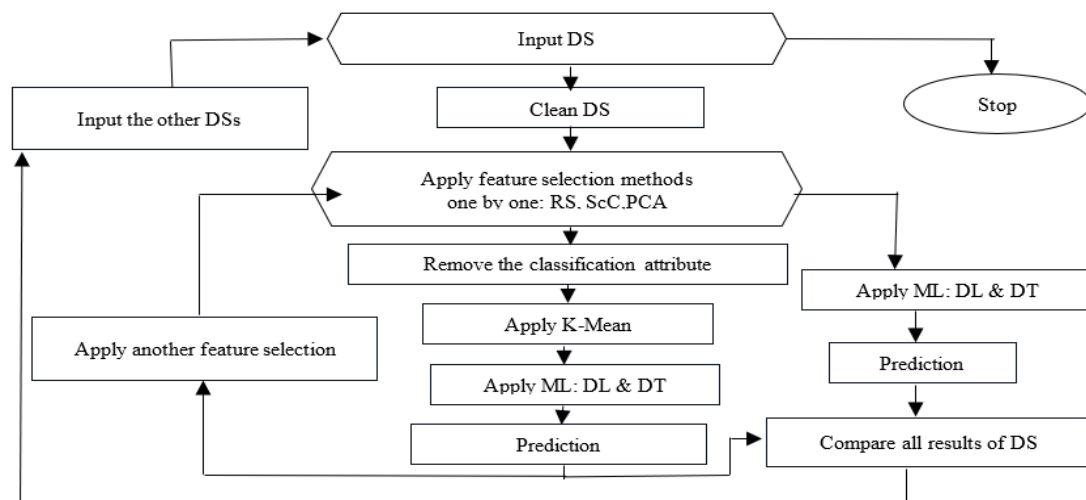


Figure 1. Algorithm of the proposed idea

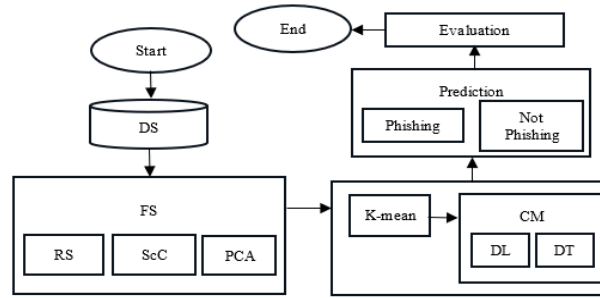


Figure 2. The components of the proposed algorithm

4.3. Simulation results

When applying feature selection methods (RS, ScC, and PCA) to the original dataset, the number of the best informative features is represented in Table 4. The performance of the models generated in Step 2 is represented in Tables 5 to 7. While the performance of the models generated in Step 6 is represented in Tables 8 to 10.

Table 4. Number of reductions in DSs

| Data set | Raw DS | RS | ScC | PCA |
|----------|--------|----|-----|-----|
| DS1 | 31 | 23 | 4 | 21 |
| DS2 | 49 | 7 | 8 | 3 |

Table 5. Measurements after applying RS to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 94.33 | 98.91 | 93.9 | 96.08 | 94.97 |
| | DT | 92.82 | 94.44 | 90.45 | 97.39 | 93.79 |
| DS2 | DL | 86.91 | 95.67 | 94.09 | 78.78 | 85.75 |
| | DT | 70.56 | 92.55 | 95.45 | 43.14 | 59.41 |

Table 6. Measurements after applying ScC to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 92.15 | 95.42 | 91.77 | 94.38 | 93.05 |
| | DT | 91.61 | 95.20 | 92.21 | 92.78 | 92.49 |
| DS2 | DL | 93.56 | 97.44 | 96.44 | 90.48 | 93.35 |
| | DT | 78.16 | 61.59 | 97.11 | 58.05 | 72.62 |

Table 7. Measurements after applying PCA to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 93.86 | 98.60 | 94.10 | 94.94 | 94.51 |
| | DT | 86.36 | 92.82 | 90.76 | 84.10 | 87.28 |
| DS2 | DL | 69.10 | 75.39 | 67.93 | 72.27 | 70.03 |
| | DT | 52.62 | 53.89 | 88.16 | 6.15 | 11.47 |

Table 8. Measurements after applying RS-K-mean to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 99.53 | 100 | 99.56 | 99.78 | 99.67 |
| | DT | 98.16 | 98.08 | 99.16 | 98.29 | 98.72 |
| DS2 | DL | 92.22 | 100 | 100 | 89.46 | 94.44 |
| | DT | 92.16 | 99.31 | 98.42 | 90.89 | 94.50 |

Table 9. Measurements after applying ScC-K-mean to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 99.37 | 100 | 98.90 | 100 | 99.45 |
| | DT | 100 | 100 | 100 | 100 | 100 |
| DS2 | DL | 96.88 | 100 | 100 | 96.31 | 98.12 |
| | DT | 97.20 | 100 | 100 | 96.70 | 98.32 |

Table 10. Measurements after applying PCA-K-mean to DSs

| Data Set | ML | Accuracy (%) | AUC (%) | P (%) | R (%) | F-m (%) |
|----------|----|--------------|---------|-------|-------|---------|
| DS1 | DL | 99.53 | 100 | 99.47 | 100 | 99.71 |
| | DT | 97.75 | 97.93 | 97.64 | 99.70 | 98.66 |
| DS2 | DL | 96.01 | 99.87 | 100 | 94.84 | 97.34 |
| | DT | 96.75 | 99.83 | 100 | 95.78 | 97.85 |

For the two reduced datasets, the accuracy of the proposed method given by DT and DL was the highest among all other feature selection methods compared to ScC, RS, and PCA. The detection accuracies of ScC-K-mean, RS-K-mean, and PCA-K-mean were significantly improved by applying the k-mean clustering method to the filter feature selection methods (ScC, RS, and PCA) using DS1 from 92.15%, 94.33%, and 93.86% to 99.37%, 99.53%, and 99.53% respectively when DL classifier was used and from 91.61%, 92.82%, and 86.36% to 100%, 98.16%, and 97.75% respectively when DT classifier was used. A comparison was also made for DS2, and the improvement was very significant.

Comparing the three suggested hybrid methods (ScC-K-mean, RS-K-mean, and PCA-K-mean), the accuracy of the ScC-K-mean using DT was the best among them with 100% accuracy for DS1 and 97.2% for DS2. For AUC and precision, the ScC-K-mean method shows higher performance (mostly 100%) for most of the readings compared to RS-K-mean, and PCA-K-mean and very high values for recall and F-measure. It is clear from the results that applying K-mean clustering after the feature selection methods (ScC-K-mean, RS-K-mean, and PCA-K-mean) achieves higher performance in terms of accuracy, AUC, precision, recall, and F-measure for the selected DSs. And among them, ScC-K-mean was the best.

K-mean, in this work, proves its performance. K-mean clustering after feature selection using RS, ScC, and PCA performs better than only feature selection (RS, ScC, and PCA), but it depends on the specific issue that being solved and the nature of data. Each of these techniques serves different purposes and excels in distinct scenarios.

The performance of the proposed approach was compared with other hybrid approaches in the previous work used in detecting phishing websites. The accuracy of the phishing hybrid approach using genetic algorithm (GA) was 91.13% [23], while it was 95.76% using features-based ANN and K-medoids clustering algorithm [24]. The study of Suleman and Awan [25] using another generating genetic algorithm (YAGGA) gave an accuracy reached 95%. Meanwhile, the study of Vrbanič *et al.* [26] using the bat algorithm and hybrid bat algorithm gave an accuracy of 96.5%. For the work that used DPI, SDN, and ANN, the accuracy was 98.39% [27]. The accuracy of the method that uses ScC and forward feature selection methods was 92.56% [31]. As our proposed method's highest accuracy for the UCI phishing websites was 100% using the DT classifier and 99.37% using the DL classifier, we proudly concluded that our approach is the pioneer in solving phishing problems.

Figures 3 (a) to 3(b) compares between measurements before and after applying the K-mean clustering for RS. Figures 4 (a) to 4(b) compares between measurements before and after applying the K-mean clustering for ScC. While Figures 5(a) to 5(b) compares between measurements before and after applying the K-mean clustering for PCA.

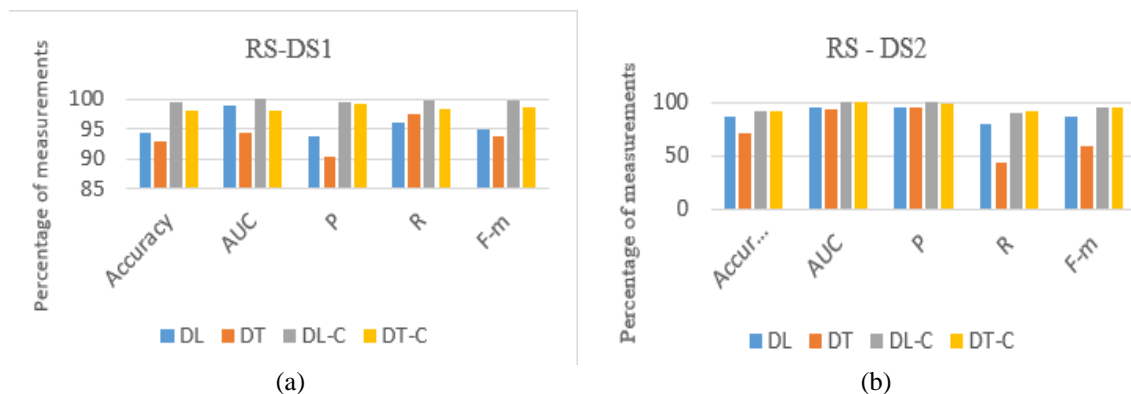


Figure 3. Comparisons between measurements before and after K-mean clustering – RS for (a) DS1 and (b) DS2

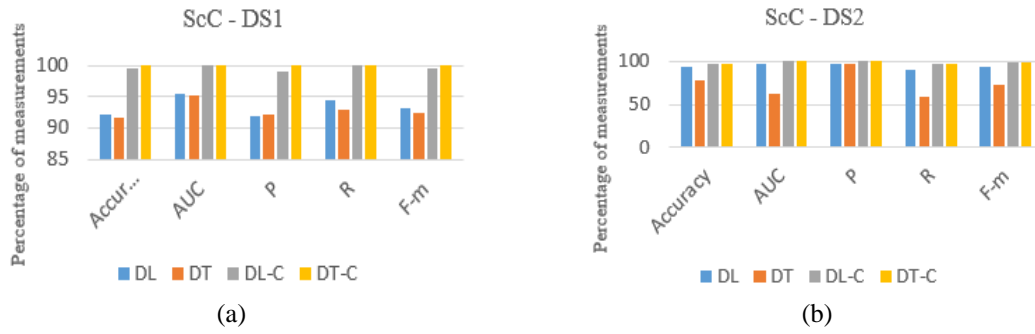


Figure 4. Comparisons between measurements before and after clustering – ScC for (a) DS1 and (b) DS2

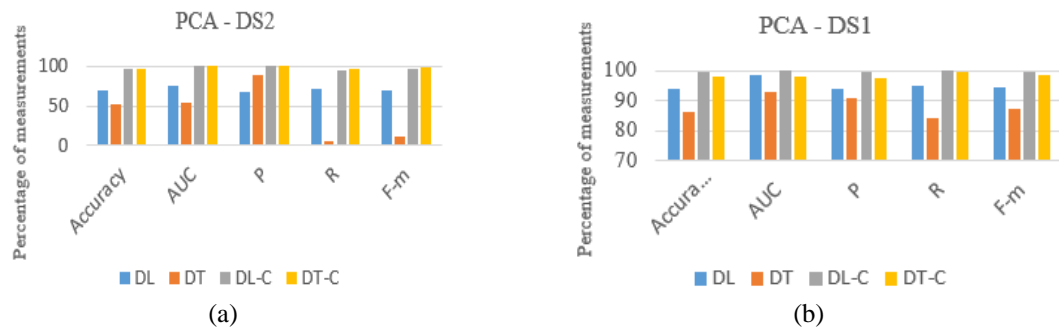


Figure 5. Comparisons between measurements before and after clustering – PCA for (a) DS1 and (b) DS2

5. COMPLEXITY OF FEATURE SELECTION AND CLUSTERING METHODS

Complexity (Big O-Notation) provides a way to compare and analyze the efficiency of algorithms and to understand how they will perform as the input size increases. The computational complexity of RS feature selection methods depends on the specific algorithm and approach being used. The complexity is typically expressed in terms of the number of instances (n) and the number of features (m) in the dataset. The complexity is typically $O(n*m^2)$ in the worst case [36]. While the computational complexity of ScC feature selection methods [37] depends on the specific algorithms and measures used for feature selection within these frameworks. Both ScC feature selection methods may involve computing correlations and stability measures for features. And the complexity of ScC feature selection typically depends on computing feature stability, which often involves calculating the Jaccard index or a similar measure for assessing feature stability across subsets of the data. The complexity is $O(n)$, where n is the number of instances.

Calculating the correlation between features and the target variable (e.g., using the Pearson correlation coefficient). The complexity of computing correlations is often $O(n*m)$, where m is the number of features. The overall complexity of stability-correlation feature selection is typically dominated by the correlation computation, which is $O(n*m)$, assuming that the stability measure is relatively efficient. Next method is the correlation-based feature selection, which focuses on computing the correlation between individual features and the target variable. The complexity is typically $O(n*m)$, where n is the number of instances, and m is the number of features. The last feature selection is PCA [38] which reduces the dimensionality of the data by creating a new set of orthogonal features called principal components. While PCA itself doesn't have a traditional computational complexity in terms of big O notation, it involves calculating eigenvectors and eigenvalues. The computational complexity of PCA mainly depends on the singular value decomposition (SVD) or eigendecomposition of the data's covariance matrix. The complexity can be expressed as $O(m^2*n) + O(m^3)$, where (m) is the number of features (original dimensions), and (n) is the number of instances (data points).

The first term, $O(m^2*n)$, represents the computational complexity of calculating the covariance matrix, and the second term, $O(m^3)$, represents the complexity of finding the eigenvectors and eigenvalues of the covariance matrix. Keep in mind that PCA is typically used to transform the data into a new space where the most important information is retained, rather than selecting a subset of the original features. The computational complexity of clustering algorithms in DL can vary widely depending on the specific clustering method, data size, and characteristics. In the case of K-Means clustering that involves iterating over the dataset

to assign data points to clusters and update cluster centroids. The time complexity for K-Means is typically $O(n \cdot k \cdot I \cdot d)$, where: n is the number of data points (instances), k is the number of clusters, I is the number of iterations, d is the number of features (dimensions).

The number of iterations (I) can vary, and typically K-Means converges relatively quickly, but it's not guaranteed to converge to a global optimum. Clustering itself does not directly affect the time complexity of feature selection methods. However, there can be indirect relationships between clustering and feature selection that may impact the overall computational complexity of a ML pipeline, such as preprocessing, feature importance, data size, and parallelization.

6. CONCLUSION

The choice of using RS, ScC, PCA, K-mean, DT, or DL depends on the specific problem, data, and goals. Each of these techniques has its strengths and weaknesses, and the right choice should be based on the characteristics of the analysis. This research proposes a hybrid method of traditional feature selection methods, K-mean clustering in addition to classification using DL and DT (ScC-K-mean) for two different DSs that include data about phishing detection. Simulation results show that the proposed algorithm outperforms the traditional tested methods that use DL and DT together with ScC, RS, and PCA methods. Also the other proposed hybrid methods that use DL and DT together with RS-K-mean and PCA-K-mean, and other hybrid methods explained earlier in section 4. Future studies will include comparing the proposed algorithm with other ML algorithms and investigating prospects for developing tools to improve the performance of the proposed algorithm. The proposed method could also be tested against further highly dimensioned phishing datasets, semi-structured and unstructured phishing datasets, and other types of attacks such as spam and malware.




REFERENCES

- [1] I. Guyon and A. Elisseeff, "an introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, pp. 1157–1182, 2003.
- [2] A. K. Das, S. Sengupta, and S. Bhattacharyya, "A group incremental feature selection for classification using rough set theory based genetic algorithm," *Applied Soft Computing Journal*, vol. 65, pp. 400–411, 2018, doi: 10.1016/j.asoc.2018.01.040.
- [3] A. K. Jain, S. Parashar, P. Katore, and I. Sharma, "PhishSKaPe: A content based approach to escape phishing attacks," *Procedia Computer Science*, vol. 171, pp. 1102–1109, 2020, doi: 10.1016/j.procs.2020.04.118.
- [4] A. Anhari, "Alexa dataset," *Kaggle*, 2023. Accessed: May 1, 2023. [Online]. Available: <https://www.kaggle.com/datasets/aanhari/alexa-dataset>
- [5] OpenPhish, "OpenPhish - Phishing Intelligence," *Open Phish*, 2021. Accessed: May 1, 2023. [Online]. Available: <https://openphish.com/>
- [6] "PhishTank | Join the fight against phishing" *Phish Tank*. Accessed: October 1, 2023. [Online]. Available: <https://www.phishtank.com/index.php>
- [7] G. Sonowal and K. S. 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, doi: 10.1016/j.jksuci.2017.07.005.
- [8] "Phishload dataset," *Phish Load*, 2023. Accessed: May 1, 2023. [Online]. Available: <https://www.medien.fh-lmu.de/team/max.maurer/files/phishload/download.html>
- [9] R. S. Rao, A. R. Pais, and P. Anand, "A heuristic technique to detect phishing websites using TWSVM classifier," *Neural Computing and Applications*, vol. 33, no. 11, pp. 5733–5752, 2021, doi: 10.1007/s00521-020-05354-z.
- [10] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Computing*, vol. 23, no. 12, pp. 4315–4327, 2019, doi: 10.1007/s00500-018-3084-2.
- [11] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Phishing websites features," *School of Computing and Engineering, University of Huddersfield*, pp. 1–7, 2015.
- [12] R. Mohammad and L. McCluskey, "Phishing websites," *UCI Machine Learning Repository*, 2012, doi: 10.24432/C51W2X.
- [13] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. 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.
- [14] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, "An adaptive machine learning based approach for phishing detection using hybrid features," *2019 5th International Conference on Web Research, ICWR 2019*, pp. 281–286, 2019, doi: 10.1109/ICWR.2019.8765265.
- [15] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decision Support Systems*, vol. 107, pp. 88–102, 2018, doi: 10.1016/j.dss.2018.01.001.
- [16] J. Nazario, "Index of /~jose/phishing," *Monkey*. Accessed: May 1, 2023. [Online]. Available: <https://monkey.org/~jose/phishing/>
- [17] "Index of /old/publiccorpus," *Spam Assassin*, 2019. Accessed: May 1, 2023. [Online]. Available: <https://spamassassin.apache.org/old/publiccorpus/>
- [18] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Computer Networks*, vol. 178, 2020, doi: 10.1016/j.comnet.2020.107275.
- [19] "Common crawl-open repository of web crawl data," *Common Crawl*. Accessed: May 01, 2023. [Online]. Available: <http://commoncrawl.org/>
- [20] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms," *SKIMA 2015 - 9th International Conference on Software, Knowledge, Information Management and Applications*, 2016, doi: 10.1109/SKIMA.2015.7399985.




- [21] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," *2017 International Conference on Electrical and Computing Technologies and Applications, ICECTA 2017*, vol. 2018, pp. 1–5, 2017, doi: 10.1109/ICECTA.2017.8252051.
- [22] "WEKA dataset," *Waikato GitHub*, 2023. Accessed: May 1, 2023. [Online]. Available: <https://waikato.github.io/weka-wiki/datasets/>
- [23] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Information Security*, vol. 13, no. 6, pp. 659–669, 2019, doi: 10.1049/iet-ifs.2019.0006.
- [24] E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, "DFOB-ANN: An artificial neural network phishing detection model based on decision tree and optimal features," *Applied Soft Computing Journal*, vol. 95, 2020, doi: 10.1016/j.asoc.2020.106505.
- [25] M. T. Suleman and S. M. Awan, "Optimization of URL-based phishing websites detection through genetic algorithms," *Automatic Control and Computer Sciences*, vol. 53, no. 4, pp. 333–341, 2019, doi: 10.3103/S0146411619040102.
- [26] G. Vrbanič, I. Fister, and V. Podgorelec, "Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification," *ACM International Conference Proceeding Series*, 2018, doi: 10.1145/3227609.3227655.
- [27] T. Chin, K. Xiong, and C. Hu, "Phishlimiter: A phishing detection and mitigation approach using software-defined networking," *IEEE Access*, vol. 6, pp. 42513–42531, 2018, doi: 10.1109/ACCESS.2018.2837889.
- [28] W. Chen, X. A. Wang, W. Zhang, and C. Xu, "Phishing detection research based on pso-bp neural network," *Advances in Internet, Data & Web Technologies*, vol. 17, pp. 990–998, 2018, doi: 10.1007/978-3-319-75928-9_91.
- [29] A. Kumar, S. S. Roy, S. Saxena, and S. S. Rawat, "Phishing detection by determining reliability factor using rough set theory," *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*, pp. 236–240, 2014, doi: 10.1109/ICMIRA.2013.51.
- [30] L. Al-Shalabi, "New feature selection algorithm based on feature stability and correlation," *IEEE Access*, vol. 10, pp. 4699–4713, 2022, doi: 10.1109/ACCESS.2022.3140209.
- [31] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R. E. Ulfath, and S. Hossain, "Phishing attacks detection using machine learning approach," *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, pp. 1173–1179, 2020, doi: 10.1109/ICSSIT48917.2020.9214225.
- [32] "UCI machine learning repository," 2017. [Online]. Available: <https://archive.ics.uci.edu>
- [33] K. Althobaiti, M. K. Wolters, N. Alsufyani, and K. Vaniea, "using clustering algorithms to automatically identify phishing campaigns," *IEEE Access*, vol. 11, pp. 96502–96513, 2023, doi: 10.1109/ACCESS.2023.3310810.
- [34] S. Mondal, D. Maheshwari, N. Pai, and A. Biwalkar, "A review on detecting phishing URLs using clustering algorithms," *2019 6th IEEE International Conference on Advances in Computing, Communication and Control*, 2019, doi: 10.1109/ICAC347590.2019.9036837.
- [35] M. Miškuf and I. Zolotová, "Comparison between multi-class classifiers and deep learning with focus on industry 4.0," *2016 Cybernetics & Informatics (K&I)*, Levoca, Slovakia, 2016, pp. 1–5, doi: 10.1109/CYBERI.2016.7438633.
- [36] S. H. Liu, Q. J. Sheng, B. Wu, Z. Z. Shi, and F. Hu, "Research on efficient algorithms for rough set methods," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 26, no. 5, pp. 524–529, 2003.
- [37] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing webpage classification via deep learning-based algorithms: An empirical study," *Applied Sciences*, vol. 11, no. 19, 2021, doi: 10.3390/app11199210.
- [38] M. Zareapoor and K. R. Seeja, "Feature extraction or feature selection for text classification: a case study on phishing email detection," *International Journal of Information Engineering and Electronic Business*, vol. 7, no. 2, pp. 60–65, 2015, doi: 10.5815/ijieeb.2015.02.08.

BIOGRAPHIES OF AUTHORS



Luai Al-Shalabi    is an Associate Professor of data mining at Arab Open University, Kuwait Branch. He completed his Ph.D. in computer science in 2000 with a focus on data mining. His areas of interest include data mining, data science, knowledge discovery, and machine learning. He has a plenty of publications in reputable local and international conferences and journals, mostly on data mining and its applications. He was a recipient of excellence award in the scientific research from the Arab Open University in Kuwait for the academic year 2019/2020. He can be contacted at email: lshalabi@aou.edu.kw.



Yahia Hasan Jazyah    received the B.S. degree in Communications and Electronics Engineering from Applied Science University, Jordan, in 2000, M.Sc. degrees in Computer Science from Amman Arab University, Jordan in 2005, and the Ph.D. degree in Data Telecommunications and Networks from the University of Salford, UK in 2011. Since 2019, he has been an associate Professor with the Information Technology and Computing, Arab Open University, Kuwait. He is the author of many journal articles and conference proceedings. His research interests include wireless routing protocols for UWB MANET, 5G, WSN, and BGP. He is an academic reviewer in several international journals. He was a recipient of excellence award in the scientific research from the Arab Open University in Kuwait for the academic year 2018/2019. He can be contacted at email: yehia_hassan@yahoo.com or yahia@aou.edu.kw.