Phishing Detection Using Hybrid Algorithm Based on Clustering and Machine Learning

Luai Al-Shalabi[1], Yahia Hasan Jazyah[2]

[1] *Faculty of Computer Studies, Kuwait, Kuwait*
[2] *Faculty of Computer Studies, Kuwait, Kuwait*

E-mail address: lshalabi@aou.edu.kw, yahia@aou.edu.kw

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: 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 behaviour. This research proposes a hybrid algorithm to detect phishing attacks based on ScC filter feature selection, clustering, and classification ML methods: Deep Learning (DL) and Decision Tree (DT). Simulation results show that the proposed technique achieves high percentage of accuracy in detecting phishing.

Keywords: AI, Decision Tree, Deep Learning, Machine Learning, Phishing.

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 revealing 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 frustrate 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 and 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 AI, 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 persistent.

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 robust hybrid algorithm for feature selection that merges between filter feature selection method and clustering using classification DL and DT for three different Datasets (DSs).

The main contributions of this work are three-fold:

- A robust hybrid feature selection method based on the ScC filter method, and the k-mean cluster method (ScC-K-mean) was proposed considering the speed and simplicity of both.
- Thorough statistical analysis of the proposed method using three well-known phishing datasets.
- A comparison between the proposed method and other well-known feature selection methods such as ScC, RS, and PCA using machine learning DT and DL methods.

The remaining of this article is organized as follows: part 2 presents phishing overview, part 3 presents phishing detection algorithms, part 4 presents comparisons between phishing detection algorithms, part 5 presents the proposed algorithm, part 6 presents the complexity of the proposed algorithm, and part 7 is the conclusion.

2. PHISHING OVERVIEW

Phishing [1] in cybersecurity refers to a malicious and deceptive practice in which cybercriminals attempt to trick individuals or organizations into disclosing sensitive and confidential information, such as login credentials, financial data, or personal information. This fraudulent activity typically, occurs through various digital communication channels, most commonly email, but also via text messages, social media messages, or fraudulent websites.

Phishers often disguise themselves as trustworthy entities, like reputable companies, government agencies, or financial institutions, to create a false sense of trust and urgency. The aim is to lure victims into clicking on links or downloading malicious attachments that, when interacted with, can lead to detrimental consequences. These can include identity theft, financial fraud, unauthorized access to accounts, or the installation of malware on the victim's device.

Phishing attacks can take different forms, including:

Email Phishing [2], where the attacker sends deceptive emails with links or attachments that, when clicked, lead to fraudulent websites or malware downloads.

Spear Phishing [3], this form of phishing is highly targeted, often focusing on specific individuals or organizations, and it may involve extensive research to craft convincing messages.

Vishing [4], which is phishing that occurs over voice calls, where scammers impersonate trusted entities to extract sensitive information.

Smishing [5], it is phishing- conducted via SMS or text messages, where victims are tricked into responding to fraudulent links or divulging personal information.

Pharming [6], where attackers redirect victims to malicious websites, even if the victim enters the correct website address, by manipulating DNS settings or using other techniques.

3. PHISHING DETECTION

Phishing detection algorithms are crucial in identifying and mitigating phishing attacks in the realm of cybersecurity. These algorithms use various techniques and methods to analyse digital content, behaviour, or network traffic to determine whether a particular instance is a phishing attempt. Below are some common algorithms and approaches used for phishing detection:

A. Rule-Based Detection

Rule-based algorithms use predefined rules or patterns to detect phishing attempts. These rules may include checking for specific keywords, suspicious URLs, or patterns in email headers or content [7].

B. Signature-Based Detection

Signature-based algorithms compare incoming data to a database of known phishing signatures. When a match is found, the system flags the content as phishing. This approach is effective against well-known and previously documented phishing attacks [8].

C. Machine Learning (ML) and AI-Based Detection:

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. Some common machine learning algorithms include [9].

1) Decision Trees (DT)

DT algorithm [10] is a supervised machine learning 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 of ensemble methods, such as Random Forests, are often employed.

2) Random Forest

Random Forest [11] is an ensemble ML algorithm that is widely used for both classification and regression tasks. It is a powerful and versatile method that combines the predictions from multiple decision trees to improve accuracy and reduce overfitting. Random Forests are particularly effective in handling complex and high-dimensional data.

They are commonly used in a wide range of applications, including image classification, text classification, fraud detection, and recommendation systems.

3) Naive Bayes

The Naive Bayes algorithm [12] is a probabilistic ML algorithm that is primarily used for classification tasks. It is based on Bayes' theorem and is particularly suited for text classification and spam email detection. Despite its simplicity, Naive Bayes often performs surprisingly well in various real-world applications, especially when dealing with large datasets and high-dimensional feature spaces.

While Naive Bayes has its simplicity and efficiency working in its favour, it may not always produce the most accurate results, especially when the independence assumption is not met.

4) Support Vector Machines (SVM)

SVMs [11] are a powerful class of supervised machine learning algorithms used for classification and regression tasks. They are widely recognized for their effectiveness in various real-world applications and their ability to handle both linear and non-linear data. However, their performance may be sensitive to the choice of the kernel function and hyperparameters, and they may not be the best choice for very large datasets.

5) Neural Networks

Deep learning neural networks [13], often referred to simply as deep neural networks or deep learning models, represent a subset of Artificial Neural Networks (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 machine learning and artificial intelligence tasks.

Deep learning neural networks, 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 labelled data for training. Proper architecture selection, hyperparameter tuning, and data preprocessing are crucial for the successful deployment of DL models.

D. Behavioural Analysis

Behavioural analysis algorithms monitor user behaviour to detect anomalies, which may indicate phishing attempts. For example, they can identify unusual login patterns or deviations from typical communication behaviour.

E. URL Analysis

Algorithms can analyse website URLs to detect inconsistencies or look for indicators that suggest a website is malicious. They may examine domain names, subdomains, and URL structure [14].

F. Content Analysis

Content analysis algorithms use Natural Language Processing (NLP) techniques to analyse the textual content of emails or websites, they are looking for deceptive language, misspellings, or other indicators of phishing [15].

G. Blacklists and Reputation-Based Approaches

These algorithms use databases of known malicious websites or email senders to identify and block phishing attempts [16].

H. Heuristic-Based Detection

Heuristic algorithms employ a set of predefined rules or heuristics to determine the likelihood of an email or website being a phishing attempt based on characteristics such as the presence of forms requesting sensitive information [18].

I. Real-Time Analysis

Phishing detection systems continuously monitor incoming data and analyse it in real-time, allowing for swift identification and prevention of phishing attempts.

Phishing detection often combines multiple methods and algorithms to enhance accuracy and effectiveness. As phishing attacks continue to evolve and become more sophisticated, these detection algorithms must adapt and improve to provide robust protection against this cybersecurity threat [18].

4. PHISHING DETECTION ALGORITHMS' COMPARISONS

Table 1 summarises the advantages and disadvantages of phishing detection algorithms.

Table 2 presents a comparison between different methods for phishing websites detection methods in terms of accuracy, which is a metric that measures how well a

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 analyse 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 A. K. et al. [19]	Modified TF-IDF	Alexa dataset [20], OpenPhish [21], Phish Tank [22]	89%
	Sonowal G. and Kuppusamy K. S. [23]	PhiDMA framework incorporates five layers	Phishload, 2016. Legitimate URL dataset [24]	92.72%
Heuristics-based	Rao R. S. et al. [25]	TWSVM	PhishTank [22], Alexa dataset [20]	98.05%
	Babagoli M. et al.[26]	meta-heuristics (HS, SVM)	UCI phishing Datasets [27], [28]	92.80%
ML	Chiew K. L. et al. [29]	Cumulative Distribution Function gradient (CDF-g), Random Forest, SVM, Naive Bayes, C4.5, JRip, and PART	UCI phishing Datasets [27], [28]	94.6%
	Yadollahi M. M. [30]	XCS	Real URLs	98.39%
DL	Smadi S. et al. [31]	Reinforcement Learning, Neural Network	PhishingCorpus [32], SpamAssassin [33], PhishTank [22]	97%
	Wei W. et al. [34]	convolutional neural networks	PhishTank [22], Common Crawl Foundation [35]	99.98%
Data Mining	Smadi S. et al. [36]	J48 algorithm and C4.5 algorithm	PhishingCorpus[32], SpamAssassin [33]	98.87%
	Subasi A. [37]	Random Forest	UCI [28], WEKA [38]	97.36%

TABLE 3: ACCURACY OF HYBRID METHODS FOR PHISHING WEBSITES DETECTION.

Anti-Phishing Method	Authors	Techniques	Dataset	Accuracy
Hybrid Methods	Ali W. and Ahmed A. A. [39]	deep neural networks (DNNs) and genetic algorithm (GA)	UCI phishing websites [28]	91.13
	Zhu E. et al. [28]	Decision Tree and Optimal Features based Artificial Neural Network, K-medoids clustering algorithm	UCI [27][28], PhishTank [22], Alexa [20]	95.76%
	Suleman M. T. and Awan S. H. [41]	Iterative Dichotomiser-3 (ID3) and Yet Another Generating Genetic Algorithm (YAGGA)	UCI machine learning website [27], [28]	95%
	Vrbančič G. et al. [42]	bat algorithm (BA) and hybrid bat algorithm (HBA)	UCI [28]	96.5%
	Chin T. et al. [43]	Deep Packet Inspection (DPI), Software-Defined Networking (SDN) and ANN	UCI[28]	98.39%
	Chen W. et al. [44]	Particle Swarm Optimization (PSO) and BP neural network	Phishtank [22]	98.95%

phishing detection system or algorithm correctly identifies and classifies phishing emails or websites.

Accuracy is calculated as a ratio represented in equation 1.

$$\text{Accuracy} = (TN+TP)/(TN+TP+FN+FP) \quad (1)$$

Where:

TP (True Positives) is the number of phishing emails or websites correctly identified as phishing.

TN (True Negatives) is the number of legitimate (non-phishing) emails or websites correctly identified as non-phishing.

FP (False Positives) is the number of legitimate emails or websites incorrectly identified as phishing (a type of error).

FN (False Negatives) is the number of phishing emails or websites incorrectly identified as legitimate (another type of error).

And Table 3 presents a comparison between different hybrid methods for phishing websites detection methods in terms of accuracy.

5. PROPOSED ALGORITHM

The proposed algorithm is a hybrid method for feature selection that merges between filter feature selection method, clustering [46] and classification ML methods using DL (H2O) [47] algorithm, which is an open-source ML platform that is designed for scalable and distributed data analysis. It has the ability to perform a wide range of ML tasks efficiently and effectively, particularly for large datasets.) and DT as represented in Fig. 1.

The goal of this study is to minimize the size of the original dataset by only keeping the most informative features. As a result, it will minimize the training and detection time as well as enhance the accuracy of the detection of phishing websites. To the best of my knowledge, it is the first study that combines both ScC and clustering. Results are greatly comparable to other methods.

First, the proposed method employs ScC method which has a comparable facility to detect the highly important features within the original dataset. Next, the classification feature of the reduced dataset generated from the previous step is omitted before passing it to the k-mean clustering method which creates two clusters (phished and not). After that, the resulting data was

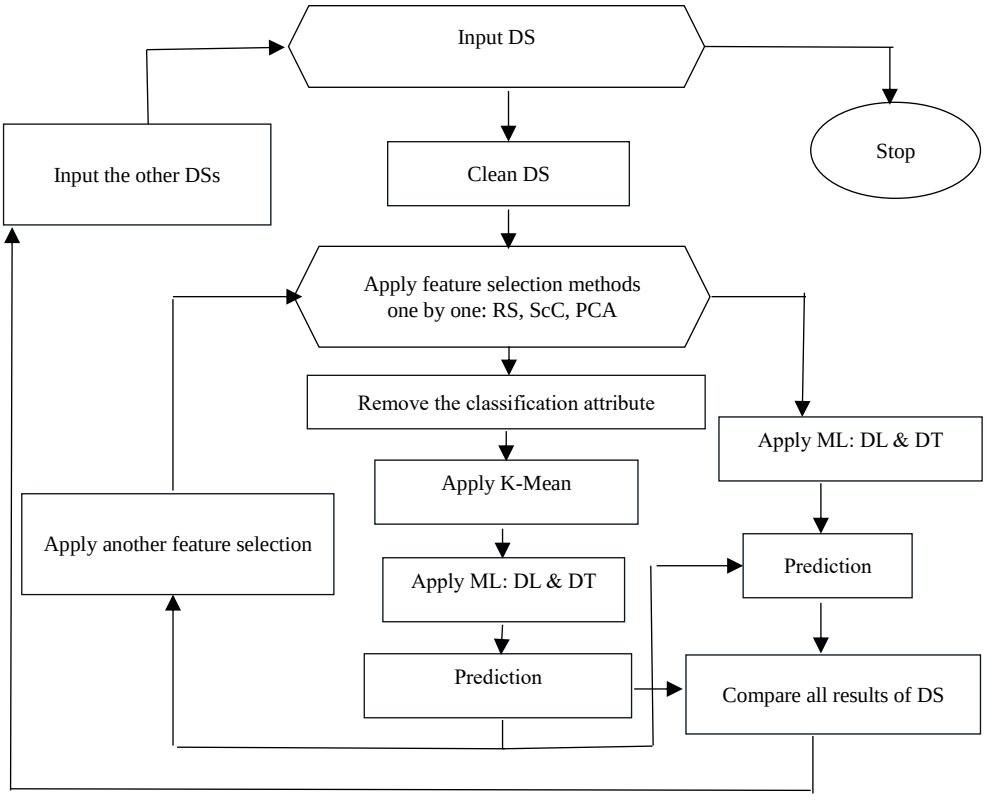


Figure 1. Algorithm of the proposed idea

formulated in a way that combines all rows of the two clusters in one dataset and creates a new classification feature based on the defined clusters. Finally, machine learning DT and DL were used for detection purposes. Evaluation metrics were used for testing the performance of the detection results. Merging both ScC and K-mean techniques has shown their potential to improve feature selection and detection performance for phishing problems. More details are shown next.

Step1. Preparing the datasets:

Three DSs [48,49,50] are used for testing, they are cleaned by removing redundancy and missing values.

Step 2. Applying feature methods:

Feature selection methods are classified into three groups: filter, wrapper, and embedded [51,52]. 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 (the most presenting of dataset which have the highest information can distinguish between classes).

The resultant classes are denoted as phishing or not phishing.

The applied methods are RS (Rough Set) [53] theory – 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 analyses the dependency of the class labels on that feature. The primary equation (equation 2) involved in rough set feature selection is the Dependency Score, which measures the importance of a feature in classifying data.

$$\text{Dependency}(SI, A) = |SI| / |U| * (|SI| - |SI|\_A) / |SI| \quad (2)$$

Where:

Dependency(SI, A) is the dependency score of feature A with respect to 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 a 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.

The next method is ScC (Stability-correlation and Correlation) [54] – 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 to other features.

ScC methods are two distinct approaches for selecting relevant features ML, the first method is Stability-Correlation 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 (SC) score is calculated using Equation 3

$$S = mode(x_i)/n \quad (3)$$

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

The second method is Correlation-Based Feature Selection which 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 equations 4, 5, and 6:

$$r = (1/(n-1))*((\sum_x \sum_y (x- \bar{x})(y-\bar{y})/(St_x St_y)) \quad (4)$$

$$St_x = \sqrt{((\sum (x-\bar{x})^2)/(n-1))} \quad (5)$$

$$St_y = \sqrt{((\sum (y-\bar{y})^2)/(n-1))} \quad (6)$$

Where:

n is the number of pairs of data used.

Σ is Sigma that represents the summation.

$\bar{x}$ is 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.

The last method is the PCA (Principal Component Analysis) [55]; which 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 EV (Explained Variance), 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.

Equation 7 presents The EV for a principal component k.

EV (PC_k) = (Eigenvalue_k) / (Total Eigenvalues) (7)

Where:

EV (PC_k) is the proportion of the total variance explained by the k-th PC (Principal Component).

Eigenvalue_k is the eigenvalue associated with the k-th principal component.

Total Eigenvalues is the sum of all eigenvalues obtained from PCA.

Step 3. Testing

Testing the first subset, where testing is applied using two types of ML algorithms - classifiers (DL and DT), in terms of:

accuracy (the result is phishing or not)

Area Under Curve (AUC), which 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-number (P), which is a measure of the accuracy of a model in correctly identifying positive instances among the instances it has classified as positive (true positive-TP), it provides information about the model's ability to avoid false positives (FP), it is calculated by equation 8.

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

Recall (R), known as Sensitivity or True Positive Rate, which is used to assess the performance of a binary classification model by measuring the model's ability to correctly identify all relevant instances from the positive class. It is calculated using equation 9.

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

F-measure (F-m), which provides a single measure of a classification model's performance by combining both precision and recall into a single score, it is calculated using equation 10.

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

Step 4. Clustering

After removing its classification attribute, the output of each method (PCA, ScC, RS) is fed to K-mean, which 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. We use k=2 since all datasets are used to distinguish between phishing and not phishing cases.

Step 5. Hybrid Approach

the previous results from step 4 are tested by means of DL and DT.

Step 6. Comparisons

all results obtained from steps 4 and 5 are compared together.

Algorithm 1 summarizes the whole process.

Algorithm 1. Proposed methodology

```
For each DSi
  Input DSi
  Clean DSi
  For each FSi
        Apply FSi
        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 FSi
```

End for
//loop until input all DSi
End for

Fig. 2. represents the flowchart of the previous algorithm.

A 10-fold cross-validation approach was employed for the training and testing stages.

When applying step 2, the results are represented in Table 4.

TABLE 4. NUMBER OF REDUCTIONS IN DSS

Data Sets	Raw DS	RS	ScC	PCA
DS1	31	23	4	21
DS2	49	7	8	3
DS3	80	6	24	7

While the number of rows in each DS is shown in table 5.

TABLE 5. NUMBER OF RECORDS IN DS

Data Set	Number or records
DS1	11050
DS2	10000
DS3	15367

when applying steps 3 and 4, the results of each DS are represented in tables 6. – 11.

TABLE 6. MEASUREMENTS AFTER APPLYING RS TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	81.30	95.37	93.56	67.76	78.59
DT	58.77	90.27	91.30	20.56	33.54

TABLE 7. MEASUREMENTS AFTER APPLYING RS-K-MEAN TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	95.49	100	100	94.14	96.98
DT	92.74	98.83	100	90.57	95.05

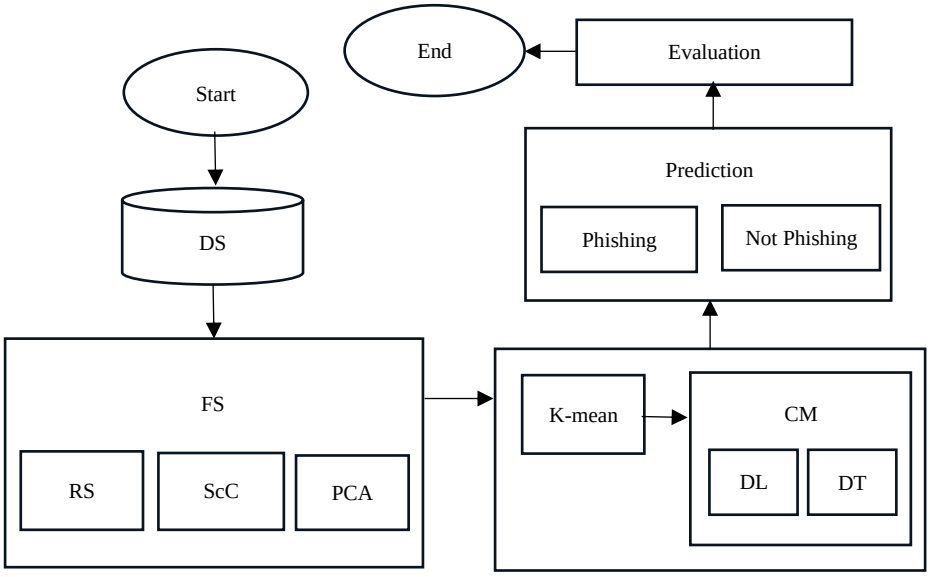


Figure 2. Flowchart of the proposed algorithm

TABLE 8. MEASUREMENTS AFTER APPLYING SCC TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	94.81	98.96	96.64	92.98	94.77
DT	76.97	93.68	93.48	58.61	72.05

TABLE 9. MEASUREMENTS AFTER APPLYING SCC-K-MEAN TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	93.92	99.94	92.62	100	96.17
DT	96.24	99.35	97.71	97.38	97.54

TABLE 10. MEASUREMENTS AFTER APPLYING PCA TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	89.20	96.77	93.93	84.12	88.75
DT	83.23	90.88	89.81	75.47	82.01

ABLE 11. MEASUREMENTS AFTER APPLYING PCA-K-MEAN TO DSS

DS1					
	Accuracy	***AUC***	***P***	***R***	***F-m***
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
DS3					
DL	96.36	99.86	100	95.35	97.62
DT	96.77	99.10	100	95.88	97.89

For all reduced datasets, the accuracy of the proposed method given by DT and DL was the highest among all other feature selection methods compared with (RS, ScC, and PCA) except for the DS3 where DL of ScC is higher by a small difference (0.89%).

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. The comparison was also made for DS2 and DS3 and the improvement was very significant (refer to figures 3-8).

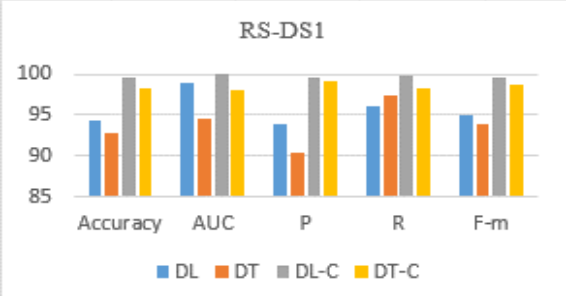


Figure 3. Comparisons between measurements before and after clustering – RS-DS1

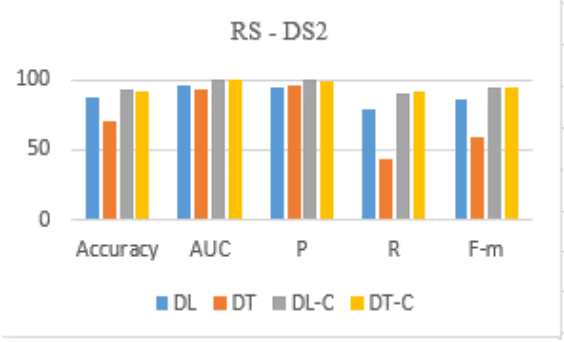


Figure 4. Comparisons between measurements before and after clustering – RS-DS2

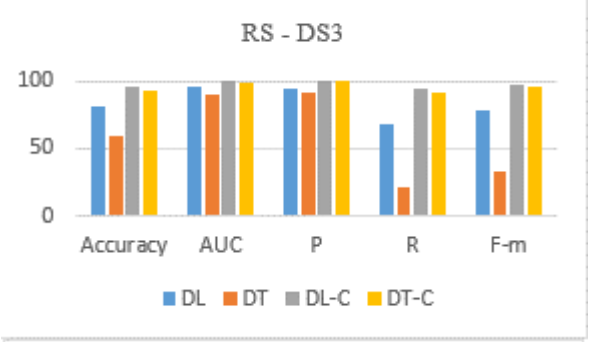


Figure 5. Comparisons between measurements before and after clustering – RS-DS3

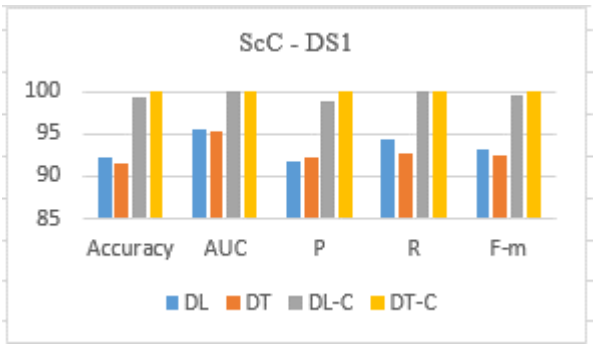


Figure 6. Comparisons between measurements before and after clustering – RS-DS3

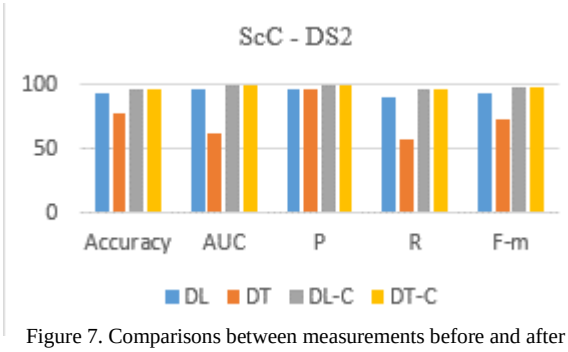


Figure 7. Comparisons between measurements before and after clustering – ScC-DS2

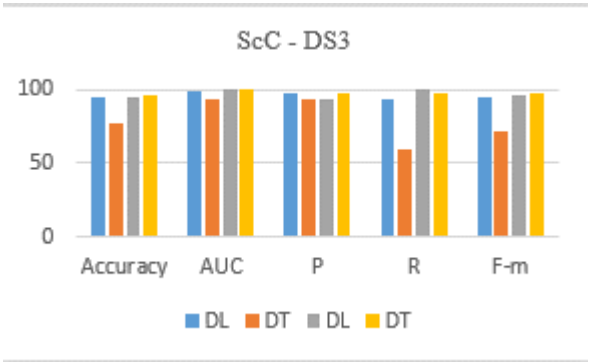


Figure 8. Comparisons between measurements before and after clustering – ScC-DS3

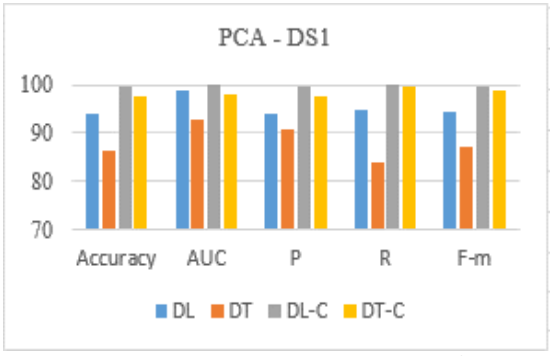


Figure 9. Comparisons between measurements before and after clustering – PCA - DS1

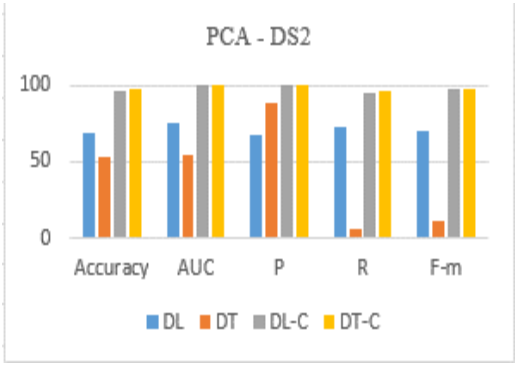


Figure 10. Comparisons between measurements before and after clustering – PCA – DS2

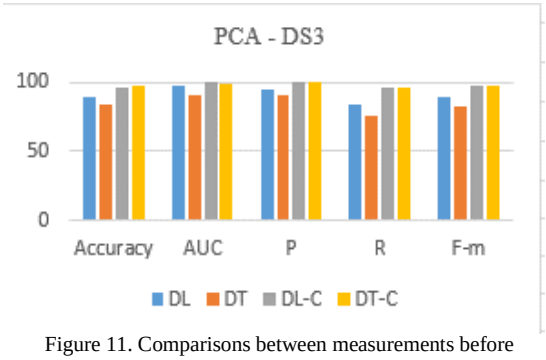


Figure 11. Comparisons between measurements before and after clustering – PCA – DS3

Focusing on our main proposed method, we tested it compared to other suggested hybrid methods (RS-K-mean and PCA-K-mean), the accuracy of the proposed method using DT was the best among them with 100% accuracy for DS1 and 97.2% for DS2 whereas it was in second place (96.24%) for DS3 with only (0.53%) difference than PCA-K-mean.

Comparing other performance metrics such as AUC and F-measure, our proposed method shows higher performance (mostly 100%) for most of the readings compared to different datasets and different machine learning classification methods. Other metrics' values for the proposed method are also high and comparable to the other methods.

It is clear from the results in Tables 6 - 11 that applying K-mean after the feature selection methods (RS,

ScC, and PCA) achieves higher performance in terms of selected DSs.

ScC-K-mean which uses clustering performs better than RS analysis, ScC analysis, 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 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 labelled 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 belong to phishing detection.

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 GA was 91.13% (Ali W. and Ahmed A. A (2019)) while it was 95.76% using features-based ANN and K-medoids clustering algorithm (Zhu E. et al. (2020)). The study of Suleman and Awan (2019) using Another Generating Genetic Algorithm (YAGGA) gave an accuracy reached 95%. Meanwhile, the study of Vrbančič G. et al. (2018) using the bat algorithm and hybrid bat algorithm gave an accuracy of 96.5%. For the work that used Deep Packet Inspection (DPI), Software-Defined Networking (SDN), and ANN, the accuracy was 98.39% (Chin T. et al. (2018)). The accuracy of the method that uses ScC and forward feature selection methods was 92.56% (Al-Shalaby, 2024). 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.

6. COMPLEXITY OF FEATURE SELECTION METHODS

Complexity (Big O-Notation) provide a way to compare and analyse 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 [56].

The computational complexity of ScC feature selection methods [57] depends on specific algorithms and measures that is used for feature selection within these frameworks. Both ScC feature selection methods may involve computing correlations and stability measures for features.

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 [58] that 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 for 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 * k * I * 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.

7. 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 own strengths and weaknesses, and the right choice should be based on the characteristics of analysis. This research proposes hybrid method of ScC feature selection and K-mean clustering in addition to classification using DL and DT for three different DSs that include data about phishing detection.

Simulation results shows that the proposed algorithm outperforms the traditional tested methods of RS, ScC, and PCA.

REFERENCES

[1] Kaur, Amandeep, and S. M. Mian. "A Review on Phishing Technique: Classification, Lifecycle and Detection Approaches." In *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 336-339. IEEE, 2023.

[2] Salloum, Said, T. Gaber, S. Vadera, and K. Shaalan. "A systematic literature review on phishing email detection using natural language processing techniques." *IEEE Access* 10 (2022): 65703-65727.

[3] Baki, Shahryar, and R. M. Verma. "Sixteen Years of Phishing User Studies: What Have We Learned?." *IEEE Transactions on Dependable and Secure Computing* 20, no. 2 (2022): 1200-1212.

[4] Yeboah-Boateng, E. Osei, and P. M. Amanor. "Phishing, SMiShing & Vishing: an assessment of threats against mobile devices." *Journal of Emerging Trends in Computing and Information Sciences* 5, no. 4 (2014): 297-307.

[5] El Karhani, Hadi, R, Al Jamal, Y. B. Samra, I. H. Elhajj, and A. Kayssi. "Phishing and Smishing Detection Using Machine Learning." In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 206-211. IEEE, 2023.

[6] Murphy, R.Diane, and R. H. Murphy. "Phishing, Pharming, and Vishing: Fraud in the Internet Age." REVIEW (2007): 37.

[7] Basnet, B. Ram, A. H. Sung, and Q. Liu. "Rule-based phishing attack detection." In *International conference on security and management (SAM 2011)*, Las Vegas, NV. 2011.

[8] Alzahrani, J. Abdullah, and A. A. Ghorbani. "Real-time signature-based detection approach for sms botnet." In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pp. 157-164. IEEE, 2015.

[9] Valiyaveedu, Nithin, S. Jamal, R. Reju, V. Murali, and K. M. Nithin. "Survey and analysis on AI based phishing detection techniques." In *2021 International Conference on Communication, Control and Information Sciences (ICCISc)*, vol. 1, pp. 1-6. IEEE, 2021.

[10] Machado, Lisa, and J. Gadge. "Phishing sites detection based on C4. 5 decision tree algorithm." In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1-5. IEEE, 2017.

[11] Noh, N. B. Md, and M. N. Bin M. Basri. "Phishing Website Detection Using Random Forest and Support Vector Machine: A Comparison." In *2021 2nd International Conference on Artificial Intelligence and Data Sciences (AIDAS)*, pp. 1-5. IEEE, 2021.

[12] Yaswanth, Palla, and V. Nagaraju. "Prediction of Phishing Sites in Network using Naive Bayes compared over Random Forest with improved Accuracy." In *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, pp. 1-5. IEEE, 2023.

[13] Bahnsen, A. Correa, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González. "Classifying phishing URLs using recurrent neural networks." In *2017 APWG symposium on electronic crime research (eCrime)*, pp. 1-8. IEEE, 2017.

[14] Charan, A. N. Soma, Y. Chen, and J. Chen. "Phishing Websites Detection using Machine Learning with URL Analysis." In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, pp. 808-812. IEEE, 2022.

[15] Pascariu, Cristian, and I. C. Bacivarov. "Detecting Phishing Websites Through Domain and Content Analysis." In *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-4. IEEE, 2021.

[16] Sinha, Sushant, M. Bailey, and F. Jahanian. "Shades of Grey: On the effectiveness of reputation-based "blacklists"." In *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 57-64. IEEE, 2008.

[17] Zuraiq, A. Abu, and M. Alkasassbeh. "Phishing detection approaches." In *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1-6. IEEE, 2019.

[18] Ahmed, A. Ali, and N. A. Abdullah. "Real time detection of phishing websites." In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1-6. IEEE, 2016.

[19] Jain, A. Kumar, S. Parashar, P. Katare, and I. Sharma. "Phishskape: A content based approach to escape phishing attacks." *Procedia Computer Science* 171 (2020): 1102-1109.

[20] "Alexa - Top sites." https://www.alexa.com/topsites (accessed May 01, 2023).

[21] "OpenPhish - Phishing Intelligence." https://openphish.com/ (accessed May 01, 2023).

[22] "PhishTank | Join the fight against phishing." https://www.phishtank.com/index.php (accessed October. 01, 2023).

[23] Sonowal, Gunikhan, and K. S. Kuppusamy. "PhiDMA–A phishing detection model with multi-filter approach." *Journal of King Saud University-Computer and Information Sciences* 32, no. 1 (2020): 99-112.

[24] “Phishload - Download.” https://www.medien.ifi.lmu.de/team/max.maurer/files/phishload/download.html (accessed May 01, 2023).

[25] Rao, R. Srinivasa, A. R. Pais, and P. Anand. "A heuristic technique to detect phishing websites using TWSVM classifier." *Neural Computing and Applications* 33 (2021): 5733-5752.

[26] Babagoli, Mehdi, M. P. Aghababa, and V. Solouk. "Heuristic nonlinear regression strategy for detecting phishing websites." *Soft Computing* 23, no. 12 (2019): 4315-4327.

[27] R. M. Mohammad, F. Thabtah, and L. McCluskey, “Phishing Websites Features,” p. 7.

[28] “UCI Machine Learning Repository: Phishing Websites Data Set.” https://archive.ics.uci.edu/ml/datasets/phishing+websites (accessed May 01, 2023).

[29] Chiew, K. Leng, 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* 484 (2019): 153-166.

[30] Yadollahi, M. Mehdi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee. "An adaptive machine learning based approach for phishing detection using hybrid features." In *2019 5th International Conference on Web Research (ICWR)*, pp. 281-286. IEEE, 2019.

[31] Smadi, Sami, N. Aslam, and L. Zhang. "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning." *Decision Support Systems* 107 (2018): 88-102.

[32] “Index of /~jose/phishing.” https://monkey.org/~jose/phishing/ (accessed May 01, 2023).

[33] “Index of /old/publiccorpus.” https://spamassassin.apache.org/old/publiccorpus/ (accessed May 01, 2023).

[34] Wei, 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* 178 (2020): 107275.

[35] “Common Crawl.” https://commoncrawl.org/ (accessed May 01, 2023).

[36] Smadi, Sami, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain. "Detection of phishing emails using data mining algorithms." In *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1-8. IEEE, 2015.

[37] Subasi, Abdulhamit, E. Molah, F. Almkallawi, and T. J. Chaudhery. "Intelligent phishing website detection using random forest classifier." In *2017 International conference on electrical and computing technologies and applications (ICECTA)*, pp. 1-5. IEEE, 2017.

[38] “WEKA.” http://www.cs.waikato.ac.nz/ml/weka/ (accessed May 01, 2023).

[39] Ali, Waleed, and A. A. Ahmed. "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting." *IET Information Security* 13, no. 6 (2019): 659-669.

[40] Zhu, Erzhou, Y. Ju, Z. Chen, F. Liu, and X. Fang. "DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features." *Applied Soft Computing* 95 (2020): 106505.

[41] Suleman, M. Taseer, and S. M. Awan. "Optimization of URL-based phishing websites detection through genetic algorithms." *Automatic Control and Computer Sciences* 53 (2019): 333-341.

[42] Vrbančič, Grega, I. F. Jr, and V. Podgorelec. "Swarm intelligence approaches for parameter setting of deep learning neural network: case study on phishing websites classification." In *Proceedings of the 8th international conference on web intelligence, mining and semantics*, pp. 1-8. 2018.

[43] Chin, Tommy, K. Xiong, and C. Hu. "Phishlimiter: A phishing detection and mitigation approach using software-defined networking." *IEEE Access* 6 (2018): 42516-42531.

[44] Chen, Wenwu, X. A. Wang, W. Zhang, and C. Xu. "Phishing detection research based on PSO-BP neural network." In *Advances in Internet, Data & Web Technologies: The 6th International Conference on Emerging Internet, Data & Web Technologies (EIDWT-2018)*, pp. 990-998. Springer International Publishing, 2018.

[45] Althobaiti, Kholoud, K. Vaniea, M. K. Wolters, and N. Alsufyani. "Using Clustering Algorithms to Automatically Identify Phishing Campaigns." *IEEE Access* (2023).

[46] Mondal, Shaheen, D. Maheshwari, N. Pai, and A. Biwalkar. "A review on detecting phishing URLs using clustering algorithms." In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pp. 1-6. IEEE, 2019.

[47] Miškuf, Martin, and I. Zolotová. "Comparison between multi-class classifiers and deep learning with focus on industry 4.0." In *2016 Cybernetics & Informatics (K&I)*, pp. 1-5. IEEE, 2016.

[48] “A 15367 benign and phishing URLs were obtained from the OpenPhish repository. Dataset”, https://openphish.com/ (accessed May 01, 2023).

[49] “Dheeru, D., Taniskidou, E.K.: UCI machine learning repository (2017)”, https://archive.ics.uci.edu/ (accessed May 01, 2023).

[50] “Tan, C.L.: Phishing dataset for machine learning: feature evaluation (2018)”, https://data.mendeley.com/datasets/h3cgnj8hft/1 (accessed May 01, 2023).

[51] Guyon, Isabelle, and A. Elisseeff. "An introduction to variable and feature selection." *Journal of machine learning research* 3, no. Mar (2003): 1157-1182.

[52] Das, K. Asit, S. Sengupta, and S. Bhattacharyya. "A group incremental feature selection for classification using rough set theory based genetic algorithm." *Applied Soft Computing* 65 (2018): 400-411.

[53] Kumar, Anugrah, S. S. Roy, S. Saxena, and S. S. Rawat. "Phishing Detection by determining reliability factor using rough set theory." *In 2013 International Conference on Machine Intelligence and Research Advancement*, pp. 236-240. IEEE, 2013.

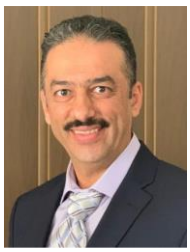
[54] L. Al-Shalabi, "New Feature Selection Algorithm Based on Feature Stability and Correlation." in *IEEE Access*, 10 (2022): 4699-4713, DOI: 10.1109/ACCESS.2022.3140209.

[55] Alam, M. Nazmul, D. Sarma, F. F. Lima, I. Saha, and S. Hossain. "Phishing attacks detection using machine learning approach." In *2020 third international conference on smart systems and inventive technology (ICSSIT)*, pp. 1173-1179. IEEE, 2020.

[56] Liu, Shao-Hui, Q. Sheng, B. Wu, Z. Shi, and F. Hu. "Research on efficient algorithms for rough set methods." *Chinese Journal of Computers-Chinese Edition*- 26, no. 5 (2003): 524-529.

[57] Do, N. Quang, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita. "Phishing webpage classification via deep learning-based algorithms: An empirical study." *Applied Sciences* 11, no. 19 (2021): 9210.

[58] Zareapoor, Masoumeh, 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* 7, no. 2 (2015): 60.



Luai Al-Shalabi is an Associate Professor of data mining at Arab Open University, Kuwait Branch. He completed his PhD 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 over 25 publications in reputable local and international conferences and journals, mostly on data mining and its applications. Dr. Al-Shalabi was a recipient of Excellence Award in the Scientific Research from the Arab Open University in Kuwait for the academic year 2019/2020.

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 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.

Dr. Yahia was a recipient of Excellence Award in the Scientific Research from the Arab Open University in Kuwait for the academic year 2018/2019.