A Robust IDS System for Intelligent Phishing Website Detection

Mosleh M. Abualhaj[1,*], Mohammad O. Hiari[1], Sumaya N. Al-Khatib[2], Ahmad Adel Abu-Shareha[3], Mohammad Sh. Daoud[4,*], Muhammad R. Faheem[5], Ali Al-Allawee[6], and Mohamad Anbar[7]
[1] Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan
[2] Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan
[3] Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan
[4] College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
[5] Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[6] Computer Science Department, University of Mosul, Mosul 41001, Iraq
[7] Cybersecuity Research Center (CYRES), Universiti Sains Malaysia (USM), Penang, Malaysia
Email: m.abualhaj@ammanu.edu.jo (M.M.A.), m.hyari@ammanu.edu.jo (M.O.H.), sumayakh@ammanu.edu.jo (S.N.A.-K.), a.abushareha@ammanu.edu.jo (A.A.A.-S.), mohammad.daoud@aau.ac.ae (M.S.D.), rehan@utem.edu.my (M.R.F.), aliabd@uomosul.edu.iq (A.A.-A.), anbar@cyres.usm.my (M.A.)

Manuscript received September 18, 2025; revised November 23, 2025; accepted December 23, 2025
*Corresponding author

***Abstract*—Phishing websites continue to pose significant threats to cybersecurity by deceiving users into revealing sensitive information. To address the complexity of high-dimensional URL features and the challenge of identifying the most informative attributes, this paper proposes a machine-learning-based phishing detection model using a hybrid Feature Selection (FS) method that combines the Dragonfly Algorithm (DA) and the Whale Optimization Algorithm (WOA). This hybrid FS approach effectively removes irrelevant attributes, reduces model complexity, and improves the robustness of the learning process. The proposed model leverages the ISCX-URL2016 dataset, with adaptive boosting (AdaBoost [AB]) serving as the classifier and its hyperparameters optimized via grid search. Experimental results show that the union of DA and WOA (DAUWOA) with AB outperforms existing methods, attaining 97.07% accuracy, 96.89% recall, 97.15% precision, 97.02% F1-score, and 94.14% Matthews Correlation Coefficient (MCC). The combination of hybrid FS and optimized classification not only boosts accuracy but also enhances computational efficiency, making the approach well-suited for real-time phishing detection systems.**

***Index Terms*—dragonfly algorithm, feature selection, machine learning, phishing, whale optimization algorithm**

I. Introduction

The web is a key and widely used technology on the Internet. The web provides a variety of services for Internet users. The wide use of the Internet has facilitated payments and financial transactions. The web is one of the most used methods for money transfer over the Internet [1, 2]. The total value of digital payments is expected to reach more than $20tn in 2025 [3]. However, the huge digital money transfer is accompanied by a huge rise in fraud [4, 5]. In the US alone, the estimated costs of cybercrime have reached more than $452bn [6]. One of the main techniques of fraud is phishing. Phishing can be accomplished through emails, messages, or even the web. Web phishing tricks Internet users into visiting fake websites. Hackers can use these fake websites to commit various types of fraud, such as stealing money or accessing sensitive information [7, 8]. Fig. 1 shows the number of detected phishing websites over the years.

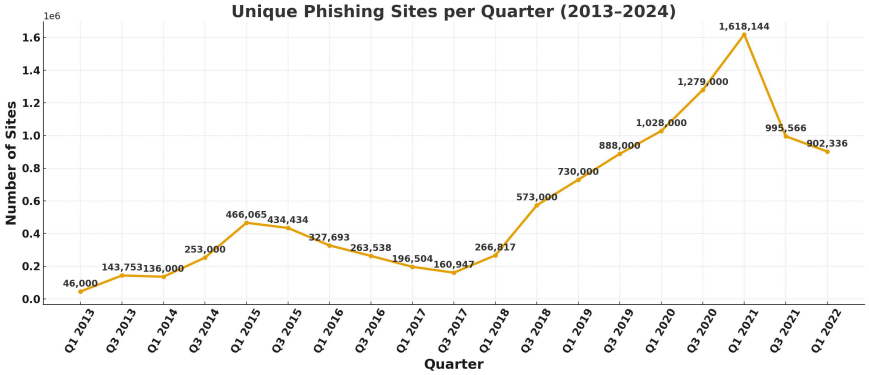

Fig. 1. Number of detected phishing websites over the years [9].

doi: 10.18178/ijeetc.15.2.116-127

Several techniques are used to detect phishing websites, including blacklists and whitelists, lexical URL analysis, and HTML & JavaScript analysis. The blacklist technique detects phishing by matching URLs against a list of previously reported malicious websites. However, the blacklist technique cannot detect new or unknown phishing websites, requires constant updating to remain effective, and is easily bypassed by slight URL modifications. The whitelist technique grants access only to URLs that appear on a trusted, pre-approved list. However, the whitelist technique blocks legitimate new websites that are not on the list, makes it difficult to maintain a comprehensive, trusted list, and is ineffective against compromised trusted sites. The lexical URL analysis technique uses URL structure, keywords, and patterns to identify suspicious or deceptive links. However, the lexical URL analysis technique may flag legitimate URLs as phishing (false positives); attackers can mimic legitimate patterns and fail against encrypted or shortened URLs. The HTML & JavaScript analysis technique inspects webpage code for malicious scripts, hidden fields, or phishing behavior. However, HTML & JavaScript analysis techniques cannot detect obfuscated or dynamically generated scripts, have high computational costs for real-time scanning, and may miss zero-day phishing exploits [10–13].

In the last decade, Machine Learning (ML) algorithms have been used extensively to detect phishing websites. There are several advantages of using ML algorithms for phishing detection. These advantages include learning from data to detect new phishing patterns, reducing reliance on manually updated blacklists, improving accuracy with continuous learning, supporting real-time detection with low latency, and automatically extracting relevant phishing indicators from data [14, 15]. ML algorithms analyze the available data and features of the phishing websites to detect them. However, ML algorithms require large labeled datasets for effective training. In addition, they are susceptible to high false alerts without good features (with the existence of irrelevant features). Moreover, real-time detection may demand high computational resources [16–18]. Therefore, Feature Selection (FS) algorithms should be used to keep only the important phishing features to detect phishing websites. The FS algorithms enhance classification accuracy by removing irrelevant features, speeding up training and prediction times of ML algorithms, reducing computational resources, and improving generalization to unseen data [16, 19].

In the last decade, Metaheuristic Optimization Algorithms (MHOA) have been widely used in FS. MHOA algorithms have been proven to be efficient in finding important features from a large set of features [20, 21]. MHOA algorithms efficiently search large feature spaces for optimal subsets, improve model accuracy by selecting informative features, handle nonlinear and complex feature interactions well, and reduce dimensionality without sacrificing predictive power. The Dragonfly Algorithm (DA) and Whale Optimization Algorithm (WOA) are two examples of the MHOA algorithms. The DA and WOA algorithms are widely used for FS in various cybersecurity domains [22–24]. In this paper, a new feature selection method that uses the union of the DA and WOA algorithms (DA∪WOA) is proposed to identify the important features of phishing websites.

Moreover, the adaptive boosting (AdaBoost [AB]) algorithm will be used to detect phishing websites based on the features selected by the DA∪WOA method. Similar to other ML algorithm, the AB algorithm is highly impacted by the choice of its hyper-parameters. Therefore, the Grid Search (GS) algorithm will be used in this study to find the values of the hyper-parameters of the AB algorithm that boosts the detection of phishing websites. Therefore, the objectives of this study are summarized as follows. First, build an ML-based phishing detection model that is able to process large phishing data. Second, a new FS method that combines the DA and WOA algorithms is proposed to identify the important features of phishing websites. Third, the AB algorithm will be used to classify phishing websites. Fourth, the hyper-parameters of the AB algorithm can be customized using the GS technique to boost the detection of phishing websites. Finally, compare the proposed ML-based phishing detection model with related works to demonstrate its efficiency.

II. URL ISCX-URL2016 Phishing Dataset

The performance of the proposed ML-based phishing detection model will be evaluated using the URL ISCX-URL2016 phishing (URL-phishing) dataset. The URL-phishing dataset is publicly available from the Canadian Institute for Cybersecurity. It is used widely in phishing detection research. The dataset contains 79 features extracted directly from the URL string and associated metadata. Table I shows the URL-Phishing dataset features. In addition, the URL-Phishing dataset contains 15367 phishing and benign samples. The URL-Phishing is considered a balanced dataset since the samples are divided into 7586 phishing samples and 7781 benign samples. Moreover, all entries in the dataset are numerical; no categorical or textual data are included [25–27].

III. Related Works

Shahrivari *et al*. [14] have compared multiple ML methods in the field of predicting phishing websites. In this context, the authors have evaluated 12 classifiers on a dataset of phishing websites, each consisting of (4898) phishing and (6157) legitimate websites. The evaluated classifiers used in this research are Support Vector Machine (SVM), logistic regression, Random Forest (RF), Decision Tree, K-nearest neighbor, AB, neural networks, gradient boosting, and XGBoost. The results collected from this work show sufficient performance using ensembling classifiers between XGBoost and RF in terms of accuracy. Several weak learners are combined to generate ensemble algorithms as a solid one. The final finding of this work is to select AB as a robust algorithm

for overfitting in low noisy datasets. Moreover, the AB algorithm is easy to visualize and understand. Some noisy data leads to poor performance due to the algorithm processing time during learning extreme cases and skewing results. In contrast to XGBoost and RF, AB is not a wise choice for fast applications to be significantly slower than XGBoost.

TABLE I: URL-PHISHING DATASET

#	Feature Name	#	Feature Name	#	Feature Name
1	Querylength	28	argDomanRatio	54	URL_sensitiveWord
2	domain_token_count	29	domainUrlRatio	55	URLQueries_variable
3	path_token_count	30	pathDomainRatio	56	spcharUrl
4	avgdomaintokenlen	31	executable	57	delimeter_Domain
5	longdomaintokenlen	32	isPortEighty	58	delimeter_path
6	tld	33	NumberofDotsinURL	59	delimeter_Count
7	charcompvowels	34	ISIpAddressInDomainName	60	NumberRate_URL
8	charcompace	35	CharacterContinuityRate	61	NumberRate_Domain
9	ldl_url	36	LongestVariableValue	62	SymbolCount_URL
10	ldl_domain	37	URL_DigitCount	63	SymbolCount_Domain
11	ldl_path	38	host_DigitCount	64	SymbolCount_Directoryname
12	ldl_filename	39	Directory_DigitCount	65	SymbolCount_FileName
13	ldl_getArg	40	File_name_DigitCount	66	SymbolCount_Extension
14	dld_url	41	Extension_DigitCount	67	SymbolCount_Afterpath
15	dld_domain	42	Query_DigitCount	68	Entropy_URL
16	dld_path	43	URL_Letter_Count	69	Entropy_Domain
17	dld_filename	44	host_letter_count	70	NumberRate_DirectoryName
18	dld_getArg	45	Directory_LetterCount	71	NumberRate_AfterPath
19	urlLen	46	Filename_LetterCount	72	Avgpathtokenlen
20	domainlength	47	Extension_LetterCount	73	Entropy_Afterpath
21	pathLength	48	Query_LetterCount	74	NumberRate_Extension
22	subDirLen	49	LongestPathTokenLength	75	Entropy_DirectoryName
23	fileNameLen	50	Domain_LongestWordLength	76	Entropy_Filename
24	this.fileExtLen	51	Path_LongestWordLength	77	Entropy_Extension
25	ArgLen	52	sub-Directory_LongestWordLength	78	NumberRate_FileName
26	pathurlRatio	53	Arguments_LongestWordLength	79	argPathRatio
27	ArgUrlRatio				

Rashid *et al*. [15] proposed an efficient ML based on the phishing detection technique. The authors utilized ML algorithms to catch the associated links of web pages and detect incoming phishing on websites. The proposed mechanism extracts features from the dataset to be compared later with the previous techniques. The algorithm is accomplished through three general steps: Step 1: the mechanism automatically collects web pages by utilizing Python and GNU Wget scripts. The whole HTML file is downloaded along with its related resources (e.g., CSS, images, JavaScript). Step 2: feature extraction is also used for host, vocabulary, and word. Step 3: Principal Component Analysis (PCA) is utilized convert large variables to small variable sets for features. The experiment was accomplished by training the split test for a comparable classification method. As a result, the experiment showed that the SVM was integrated with the classification model to achieve better performance and accurately classify 95.66% of phishing.

Yi [28] proposed a phishing detection model utilizing Deep Belief Networks (DBN) and introduced two types of features: original features and interaction features. The model was trained on a small dataset to optimize the parameters and later evaluated using real IP flow data from an Internet Service Provider (ISP). Experimental results demonstrated that the DBN-based model achieved a True Positive Rate (TPR) of approximately 90%, a False Positive Rate (FPR) of 0.6%, and accuracy of 89.6%. The findings suggest that deep learning architectures, particularly DBN, can enhance phishing detection by leveraging advanced feature representations and improving classification performance on large-scale datasets.

Alqahtani and Abu-Khadrah [29] proposed a phishing detection model that integrates three machine learning classifiers: RF, SVM, and Bagging. Their approach was trained on a dataset from the UCI Machine Learning Repository, consisting of 1,353 URLs categorized as legitimate, phishing, or suspicious. To handle data imbalance, they applied the Synthetic Minority Oversampling Technique (SMOTE), and used a correlation-based filter method for FS. This method identifies the most relevant features by examining their linear relationship with the target variable. The combined model achieved 92.33% accuracy, with precision, recall, and F1-score all exceeding 92%, outperforming individual classifiers. Their findings highlight the effectiveness of ensemble learning and careful data preprocessing in phishing detection.

Daniel *et al*. [30] proposed an effective phishing detection model by combining ML with the FS techniques. Their study evaluated the performance of RF and Artificial Neural Network (ANN) models when integrated with PCA and Recursive Feature Elimination (RFE). Using a dataset containing 4,898 phishing and 6,157 legitimate websites sourced from Kaggle, the authors demonstrated that the RF+PCA combination achieved 95.83% accuracy, while ANN+PCA reached 95.07%. The inclusion of PCA and RFE not only improved predictive performance but also enhanced

computational efficiency and reduced overfitting.

Nagy *et al*. [31] investigated the use of parallel processing techniques to enhance the efficiency of ML and Deep Learning (DL) models for phishing website detection. Using a dataset comprising 54,000 training and 12,000 testing URLs, they evaluated four models—RF, Naïve Bayes (NB), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM)—across sequential and four parallel execution settings, including multithreading and multiprocessing with Python. Their results showed significant reductions in training time, with the highest speedup of 3.51×(71.54%) achieved for LSTM using Python backend threading with two jobs. Importantly, these performance gains did not compromise detection accuracy; NB achieved 96.01% accuracy, while RF, CNN, and LSTM achieved 100% recall. The study demonstrates that applying parallel processing can enhance phishing detection systems by accelerating training time without loss of performance, offering a promising direction for real-time cybersecurity applications.

Prabakaran *et al*. [32] proposed a DL–based framework for detecting phishing attacks. The authors presented a hybrid DL–based phishing detection framework that integrates a Variational Autoencoder (VAE) with a Deep Neural Network (DNN). In this architecture, URLs are transformed into fixed-size one-hot encoded matrices (116 × 84) to ensure a uniform input structure, while the VAE functions as a feature extractor by encoding the raw URL strings into compact latent representations. The study highlights the critical role of latent space tuning in enhancing model performance. By refining these latent representations, the approach improves classification accuracy while simultaneously reducing feature dimensionality and mitigating noise. A latent dimensionality of 24 yielded the lowest reconstruction loss during VAE training. Subsequently, the DNN operates as the classification module, determining whether each URL is malicious or legitimate. The study makes use of the publicly available ISCX-URL2016 phishing dataset, applying an 80/20 split for training and testing. The proposed model achieved an accuracy of 97.45%, outperforming several benchmark methods, including convolutional neural networks (93.18%), vanilla autoencoders (94.73%), sparse autoencoders (95.09%), and denoising autoencoders (95.71%).

Vinayakumar *et al*. [33] introduced a DL-based framework designed to detect malicious phishing URLs that evade traditional blacklist-based detection systems. Their proposed algorithm, termed DeepURLDetect (DUD), employs a hybrid DL architecture that integrates CNN with LSTM networks. The model encodes raw URLs at the character level using Keras-based embeddings, enabling it to preserve sequential patterns within the URL structure while eliminating the need for manual feature engineering. The CNN component extracts local patterns—such as substrings and common phishing-related tokens (e.g., login, paypal)—while the LSTM layer captures long-term dependencies and sequential relationships among URL characters. The DUD model employs a sigmoid activation function for binary classification (malicious vs. benign) and is trained using the Adam optimizer, which minimizes binary cross-entropy loss to enable adaptive learning. The model achieved accuracies of 97.2% on Dataset 1 (D1), 95.4% on Dataset 2 with random splitting (D2R), and 93.1% on Dataset 2 with time-based splitting (D2T). The 3-gram DNN baseline model, comprising five layers, also demonstrated strong performance, achieving accuracies of 95.4%, 95.0%, and 93.0% on D1, D2R, and D2T, respectively.

IV. Proposed ML-Based Phishing Detection Model

This section discusses the operations performed by the proposed ML-based phishing detection model, shown in Fig. 2. First, the URL ISCX-URL2016 phishing dataset is imported into the model. Then, the data preprocessing operation is performed to ensure compatibility with ML algorithms, i.e., the AB algorithm. After that, the FS operation is performed to choose the most relevant features to improve model performance. Finally, the model is evaluated using the AB algorithm.

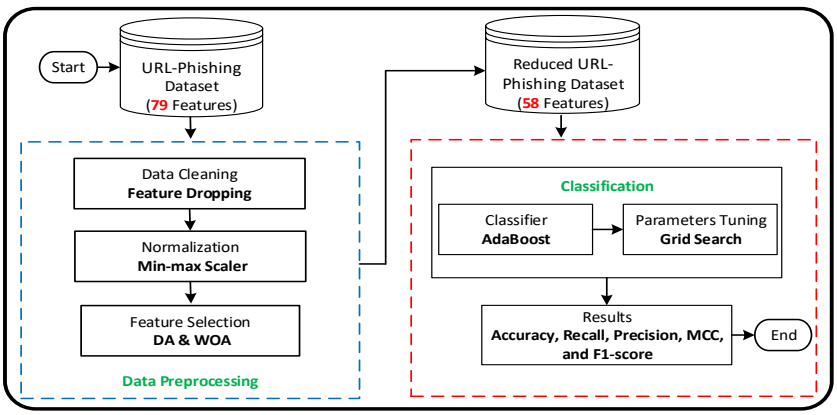

Fig. 2. the ML-based phishing detection model.

A. URL-Phishing Dataset Preparation

As mentioned earlier, the URL-Phishing dataset consists of 79 features that are used to identify a phishing attack. However, ten features in the URL-Phishing dataset contain noise data such as NaN, infinity, or even

no values. These features are 70) NumberRate_DirectoryName, 71) NumberRate_AfterPath, 72) Avgpathtokenlen, 73) Entropy_Afterpath, 74) NumberRate_Extension, 75) Entropy_DirectoryNam, 76) Entropy_Filename, 77) Entropy_Extension, 78) NumberRate_FileName, and 79) argPathRatio. We have removed these features to avoid the negative impact on phishing attack detection. Therefore, the URL-Phishing dataset is reduced to 69 features. The values of the remaining features are distributed between very small values and very large values. Therefore, the ML algorithms might be tricked and give incorrect results. We have narrowed these values to be between 0 and 1 using the min-max algorithm [34, 35]. For example, some of the values before applying the min-max algorithm are 64, 68, 69, 93, 68, 62, 98, 53, 114, and 63. These values have been replaced by 0.08681672, 0.099678457, 0.102893891, 0.180064309, 0.099678457, 0.080385852, 0.196141479, 0.051446945, and 0.247588424, respectively, after applying the min-max algorithm.

B. FS Using the DA and WOA Algorithms

The FS process reduces computational complexity, improves classification accuracy, and minimizes overfitting by eliminating redundant or irrelevant features. FS for phishing attack detection enhances model performance by identifying the most relevant attributes that distinguish phishing from legitimate instances [16, 19]. Effective FS methods, such as DA and WOA, optimize detection by focusing on critical phishing attributes like URL structure, domain age, and HTTPS usage.

1) FS using the DA algorithm

The DA Algorithm 1 mimics static and dynamic swarming behaviors of dragonflies. It balances exploration and exploitation during the search process. Inspired by Levy flight and dragonfly movement in nature, it is suitable for solving continuous and discrete optimization problems. The algorithm employs five main behaviors: separation, alignment, cohesion, attraction, and distraction. It uses a food source and an enemy to guide the search direction. As a population-based metaheuristic, it features adaptive velocity updates. It converges efficiently in complex, high-dimensional search spaces and is often hybridized with other algorithms for improved performance. The DA algorithm is especially effective in FS and ML tasks. Table II shows the main merits of the DA algorithm [24, 36]. Algorithm 1 displays the pseudocode for the DA algorithm. The DA has selected 46 features from 69 features as the key features to identify the phishing attack. These features are 4, 6, 7, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 23, 26, 27, 28, 29, 30, 31, 32, 34, 36, 37, 38, 39, 41, 42, 44, 45, 46, 50, 51, 52, 53, 54, 56, 60, 62, 63, 65, 66, 67, 68, and 69.

TABLE II: MAIN MERITS OF THE DA ALGORITHM

Aspect	Description	Operation in FS
Inspiration	Mimics dragonfly swarm behavior, including static and dynamic movements.	Explores and exploits feature space efficiently for optimal selection.
Exploration vs. Exploitation	Balances search strategies to avoid local optima and improve accuracy.	Uses attraction, repulsion, and alignment forces for diverse searching.
Search Strategy	Uses random and guided movements to explore feature subsets.	Identifies optimal phishing-related features by evaluating relevance.
FS Strengths	Effectively eliminates redundant and irrelevant features.	Reduces computational complexity and improves model efficiency.
Suitability for Phishing Detection	Enhances phishing classification performance by refining input features.	Ensures only the most informative attributes are used for detection.

Algorithm 1: Dragonfly Algorithm (DA)

psn is position, vel is velocity, dfis dragonfly

```
Initialize a population of df randomly (psn and vel)
Calculate the fitness of each df
Set the global best (gbest) as the df with the best fitness
Repeat until the maximum number ofiterations is reached:
    Update the inertia weight w (usually linearly decreasing)
    Update the separation (S), alignment (A), cohesion (C),
    attraction to food (F), and distraction from enemy (E) for each df:
        -S= avoid crowding (keep distance from neighbors)
        - A= match velocity with neighbors
        -C= move towards the center of neighbors
        - F = move towards food source (best solution so far)
        - E = move away from enemy (worst solution so far)
    For each df:
        - Calculate the step vector using:
        step=S+A+C+F+E
        - Update velocity using:
        vel=w* vel + step
        - Update position using:
        psn =psn+vel
        - Apply boundary constraints if df exceeds bounds
    Evaluate the fitness of the updated positions
    Update the global best solution if a df has better fitness
Return the best solution found
```

2) FS using the WOA algorithm

The WOA Algorithm 2 is inspired by the bubble-net hunting behavior of humpback whales. It balances exploration and exploitation in the search space effectively. The algorithm mimics whales' strategies of encircling and attacking prey. It is simple to implement and requires only a few control parameters. WOA is suitable for solving complex global optimization problems. It employs a spiral updating position mechanism along with a shrinking encircling method. The algorithm is often applied in FS and ML tasks. WOA converges quickly by efficiently exploring the search paths. It is also robust against getting trapped in local optima, especially in high-dimensional problems. Table III shows the main merits of the WOA algorithm [37, 38]. Algorithm 2 displays the pseudocode for the WOA algorithm. The WOA has selected 43 features from 69 features as the key features to identify the phishing attack. These features are 1, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 27, 28, 29, 31, 32, 33, 34, 35, 36, 38, 41, 43, 44, 45,

46, 47, 49, 50, 51, 53, 55, 56, 57, 60, 61, 63, 65, 66, 68, and 69.

TABLE III: MAIN MERITS OF THE WOA ALGORITHM

Aspect	Description	Operation in FS
Inspiration	Mimics humpback whales' bubble-net hunting strategy.	Guides FS using spiral and encircling search behavior.
Exploration vs. Exploitation	Balances global search and local refinement through adaptive mechanisms.	Prevents premature convergence while searching for optimal feature subsets.
Search Strategy	Alternates between encircling prey and spiral position updates.	Efficiently navigates the feature space to find optimal combinations.
FS Strengths	Eliminates irrelevant or redundant features with minimal tuning effort.	Enhances model performance and reduces overfitting risk.
Suitability for Phishing Detection	Improves phishing detection accuracy by selecting informative features only.	Focuses classification on the most predictive phishing-related attributes.

Algorithm 2: Whale Optimization Algorithm (WOA)

Initialize a population of whales (wh) with random positions (psn).
Evaluate the fitness of each whale.
Set the best solution (globalBest) as the whale with the highest fitness.
Repeat until the maximum number of iterations is reached:
  Update parameter'a', decreasing linearly from 2 to 0.
  For each whale in the population:
    Generate random numbers x1 and x2 between 0 and 1.
    If x2 < 0.5:
      - Update the whale's position using:
      wh.psn=globalBest.psn-(x1* distanceToGlobalBest)
    Else ifx2 >= 0.5 and lal < 1:
      Select a random whale (wh_j).
      Update the position using:
      wh.psn = wh_j-psn -(x1 * distanceToWhale_j)
    Else ifx2 >=0.5 and lal >= 1:
      Generate a random number x3 between 0 and 1.
      Update the position using:
      wh.psn =(globalBest.psn - wh.psn) * (x3 * a)
    Ensure the updated position meets problem constraints.
  Evaluate the fitness of each whale with the new position.
  Update globalBest if a better fitness is found.
Return the best solution (globalBest).

3) Proposed DA∪WOA FS method

The proposed DA∪WOA union-based feature selection method leverages the complementary strengths of two independent algorithms rather than focusing solely on aggressive dimensionality reduction. Although DA and WOA each identify informative but partially overlapping features, relying on a single subset would remove discriminative information important for phishing detection. By merging both subsets, the union approach captures broader feature coverage and patterns that neither method identifies alone, producing a richer and more descriptive representation of URL characteristics. While the final subset (58 features) is larger than the individual outputs of DA or WOA, it remains smaller than the original feature set and consistently yields higher accuracy, stronger recall, and more stable performance. Importantly, the hybrid method maintains training, inference, and execution times comparable to the individual algorithms, and its linear complexity ensures minimal overhead—a point demonstrated in detail in Section VI (Runtime Efficiency).

The proposed DA∪WOA feature-selection method combines the outputs of two independent optimization algorithms through a simple but effective union process. The procedure begins by running the DA algorithm on the full URL-Phishing dataset. DA searches the feature space according to its swarm-based movement rules and returns a subset of features it considers most informative; in our case, DA selected 46 features. In parallel, the WOA algorithm is applied to the same dataset using its own search strategy, and it independently produces a second subset of 43 selected features. Because DA and WOA rely on different exploration–exploitation mechanisms, their selections are not identical and only partially overlap.

Once both algorithms complete execution, their outputs are combined using a standard set-union operation. This step merges the two feature lists and automatically removes any duplicate indices, ensuring that each feature appears only once in the final set. The resulting union contains 58 unique features—more comprehensive than either subset alone but still smaller than the original 69 features. These features are 1, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 26, 27, 28, 29, 31, 30, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 60, 61, 62, 63, 65, 66, 67, 68, and 69. This final feature set represents the collective strengths of both algorithms, preserving informative features that might be missed if only one method were used. The FS process is illustrated in Fig. 3. The reduced feature set enhances the performance of the proposed ML-based phishing detection model. Performance results are discussed in the following sections.

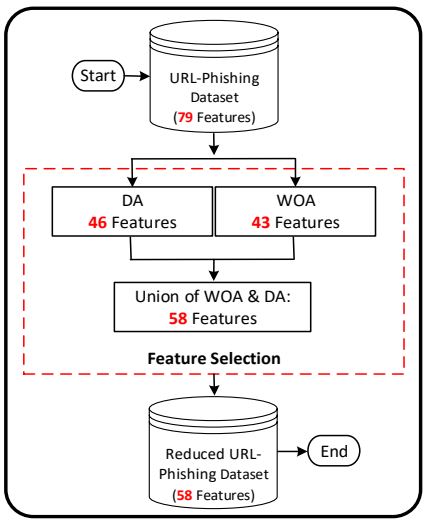

Fig. 3. Feature selection process.

C. AB Classifier

AB is a powerful ensemble method that combines

multiple weak learners into a strong classifier. It focuses on misclassified samples in successive iterations to improve accuracy. The algorithm adjusts weights to emphasize difficult training instances in each round. It commonly uses decision stumps as base learners for simplicity and speed. AB helps reduce both bias and variance in classification problems. It performs well with clean, noise-free datasets but is sensitive to noisy data and outliers. The method works effectively with both binary and multiclass classification tasks. By boosting the performance of weak learners, it improves overall generalization. AB is widely used in applications such as text classification, spam detection, and phishing website identification. The AB algorithm performs several steps, as follows to complete a classification task [36–38]:

Step 1: Initialize equal sample weights for all training instances.

Step 2: Train a weak learner using weighted samples, emphasizing higher-weighted ones.

Step 3: Calculate the weighted classification error of the learner.

Step 4: Increase weights of misclassified samples; decrease weights of correctly classified ones.

Step 5: Compute learner importance based on its error rate.

Step 6: Repeat training with updated weights to build an ensemble.

Step 7: Combine weak learners via weighted majority voting by accuracy.

Step 8: Output the final strong classifier as the weighted combination of all learners.

Similar to other classifiers, the AB classifier depends on several hyper-parameters. Fine-tuning of these hyper-parameters is highly impactful on the AB performance. Proper tuning improves model accuracy by optimizing weak learner combinations, prevents overfitting by controlling model complexity and learning speed, enhances generalization by selecting suitable learning rate, and balances the bias-variance tradeoff through number of estimators. Table IV lists the key hyper-parameters of the AB classifier [39–41]. The Grid Search (GS) algorithm can be used to tune these hyper-parameters.

GS algorithm tests all possible combinations of parameter values. It evaluates model performance for each parameter set and identifies the parameter set with the highest validation score. Typically, it is used with k-fold cross-validation to reduce overfitting. However, it can be computationally expensive for large search spaces. GS always returns the same result for the same input, and it helps find optimal parameters for the best model performance. The GS algorithm performs several steps for hyperparameter tuning. The GS algorithm specifies all hyperparameters and their possible values, generates all possible parameter combinations, splits the data using cross-validation into folds, trains the model on each parameter combination and fold, evaluates the performance by measuring accuracy or error for each fold, computes the average performance across all validation folds, and selects the parameter set with the highest average score [42–44]. The optimal hyperparameter values of the AB classifier obtained using the GS algorithm with five-fold cross-validation are presented in Table IV.

TABLE IV: KEY HYPER-PARAMETERS OF THE AB CLASSIFIER

Hyperparameter	Value	Description
n_estimators	50	Number of boosting rounds or weak learners to train.
learning_rate	1.0	Controls the weight of each learner's contribution.
base_estimator	DecisionTreeClassifier(max_depth=1)	Specifies the weak learner model to use.
algorithm	'SAMME.R'	Defines the boosting method: 'SAMME' or 'SAMME.R'.
min_samples_split	2	Minimum samples to split an internal node.
min_samples_leaf	1	Minimum samples required at a leaf node.
criterion	'gini'	Function to measure split quality (e.g., 'gini', 'entropy').
splitter	'best'	Strategy used to choose split at each tree node.

V. RESULTS AND DISCUSSION

The evaluation tests have been implemented using Python on a Lenovo IdeaPad Slim 3 15IRU8 Laptop-Intel Core i3-1315U Up to 4.50 GHz Processor (10MB Cache), 8GB DDR5, 256GB SSD M.2, Intel UHD Graphics, and Ubuntu 18.04.6 Linux operating system. The needed libraries (e.g., NumPy and TensorFlow) in Python have been installed and configured. The confusion matrix elements were used to find the value of the five different metrics that were used in the assessment, namely accuracy, precision, recall, F1-Score, and Matthews Correlation Coefficient (MCC). Table V presents the assessment metrics [45–48]. The confusion matrix elements are True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). The parameters for both DA and WOA were set to a population size of 30 and a maximum of 200 iterations. This configuration provides sufficient diversity for exploration and enough iterations for each algorithm to refine its solutions.

TABLE V: THE ASSESSMENT METRICS

Metric	Definition	Importance in Phishing Detection	Strengths	Weaknesses
Accuracy	Measures the proportion of correctly classified phishing and legitimate emails (TP + TN) / (TP + TN + FP + FN)	Gives a general performance overview of the detection model	Simple and widely used metric	Can be misleading if the dataset is imbalanced (e.g., more legitimate emails than phishing)
Precision	Measures the proportion of correctly identified phishing emails out of all emails classified as phishing TP / (TP + FP)	Evaluates how many detected phishing emails are actually phishing	Useful when false positives (FP) must be minimized to avoid flagging legitimate emails	May not be suitable if false negatives (FN) are costly (i.e., missing real phishing attacks)

Recall	Measures the proportion of actual phishing emails that were correctly identified TP / (TP + FN)	Important for ensuring phishing emails are not missed	Helps detect a higher number of phishing emails, reducing security risks	High recall can lead to an increase in false positives (FP), flagging more legitimate emails
F1-Score	Harmonic mean of precision and recall 2 × (Precision × Recall) / (Precision + Recall)	Balances precision and recall for phishing detection	Suitable for imbalanced datasets where both FP and FN are crucial	Does not provide a complete picture if precision and recall values differ significantly
MCC	Measures overall classification quality, considering all TP, TN, FP, and FN	More reliable in imbalanced phishing datasets	Provides a balanced assessment even with uneven class distributions	Less commonly used and harder to interpret compared to accuracy

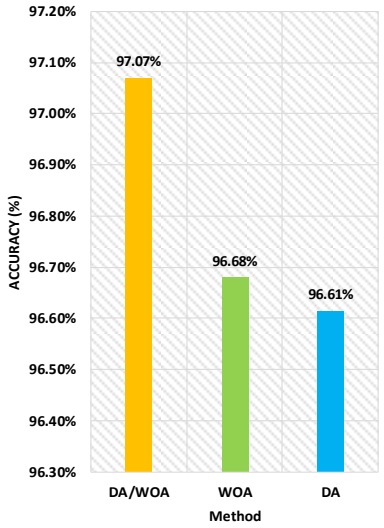

Fig. 4. Accuracy of the proposed phishing websites detection model.

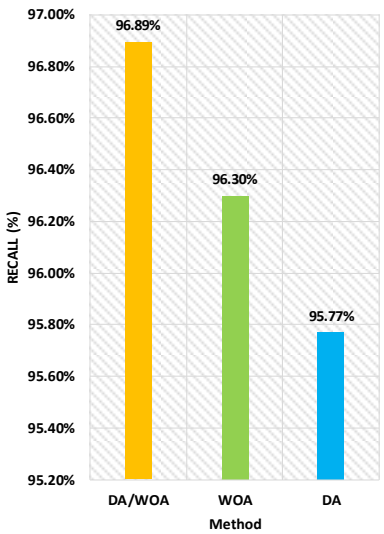

Fig. 5. Recall of the proposed phishing websites detection model.

Fig. 4 compares the accuracy of the DAUWOA, WOA, and DA methods for FS. The accuracy of the DAUWOA method is 97.07%, while WOA and DA achieve 96.68% and 96.61%, respectively. All methods achieved strong performance in terms of accuracy. However, the DAUWOA hybrid method outperforms the individual WOA and DA approaches. It achieves a higher accuracy by 0.39% over WOA and 0.46% over DA, demonstrating the benefit of combining the two optimization algorithms. This high-level of accuracy suggests the proposed method is reliable for detecting phishing attacks with minimal misclassification.

Fig. 5 compares the recall of the DAUWOA, WOA, and DA methods for FS. The recall of the DAUWOA method is 96.89%, while WOA and DA achieve 96.30% and 95.77%, respectively. All methods showed strong performance in terms of recall. However, the DAUWOA hybrid method performs better than both individual methods, with an improvement of 0.59% over WOA and 1.12% over DA. This high-recall score suggests the proposed method is capable of detecting most phishing attacks with minimal false negatives.

Fig. 6 compares the precision of the DAUWOA, WOA, and DA methods for FS. The precision of the DAUWOA method is 97.15%, while WOA and DA achieve 96.94% and 97.31%, respectively. All methods demonstrated strong performance in terms of precision. Although DA achieved the highest precision, the DAUWOA hybrid maintains a balanced and competitive result, outperforming WOA by 0.21%. This indicates that the proposed method is effective in minimizing false positives when detecting phishing attacks.

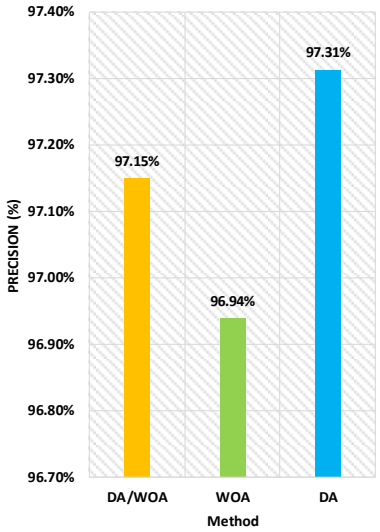

Fig. 6. Precision of the proposed phishing websites detection model.

Fig. 7 compares the F1-score of the DAUWOA, WOA, and DA methods for FS. The F1-score of the DAUWOA method is 97.02%, while WOA and DA achieve 96.62% and 96.54%, respectively. All methods showed strong performance in terms of F1-score. However, the DAUWOA hybrid method performs better than both individual methods, with an improvement of 0.40% over WOA and 0.48% over DA. This high F1-score indicates that the proposed method achieves a strong balance between precision and recall for phishing detection.

Fig. 8 compares the MCC of the DAUWOA, WOA, and DA methods for FS. The MCC of the DAUWOA method is 94.14%, while WOA and DA achieve 93.36% and 93.24%, respectively. All methods showed strong

performance in terms of MCC. However, the DAUWOA hybrid method performs better than both individual methods, with an improvement of 0.78% over WOA and 0.90% over DA. This high MCC value indicates that the proposed method maintains a strong balance between precision and recall in phishing detection.

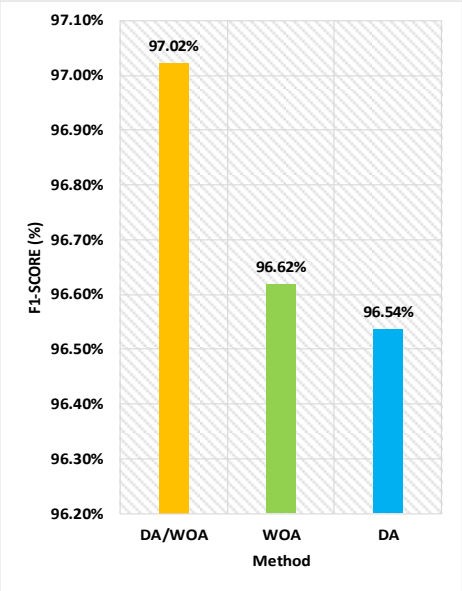

Fig. 7. F1-Score of the proposed phishing websites detection model.

The proposed phishing websites detection model (AB-DAUWOA) has been evaluated against other phishing detection models. Fig. 9 shows the accuracy of the AB-DAUWOA model in comparison to these models. The proposed AB-DAUWOA model achieved the highest accuracy at 97.07%. In contrast, the AB-DA and AB-WOA variants achieved 96.61% and 96.68%, which are 0.46% and 0.39% lower, respectively. Previous models—Rashid *et al*. (95.66%) [15], Yi *et al*. (89.60%) [28], Alqahtani and Abu-Khadrah (92.33%) [29], Daniel *et al*. (95.83%) [30], Nagy *et al*. (96.01%) [31], Prabakaran *et al*. [32] 97.33%, and Vinayakumar *et al*. [33]—show lower performance by margins of 1.41%, 7.47%, 4.74%, 1.24%, 1.06%, -0.26%, and -0.13%, respectively.

These results demonstrate the superiority of the proposed AB-DAUWOA model in phishing detection. By combining the merits of DA and WOA algorithms, the hybrid FS enhances classification accuracy and computational efficiency. This leads to better model performance and highlights the effectiveness of the proposed method in selecting informative features and detecting phishing threats.

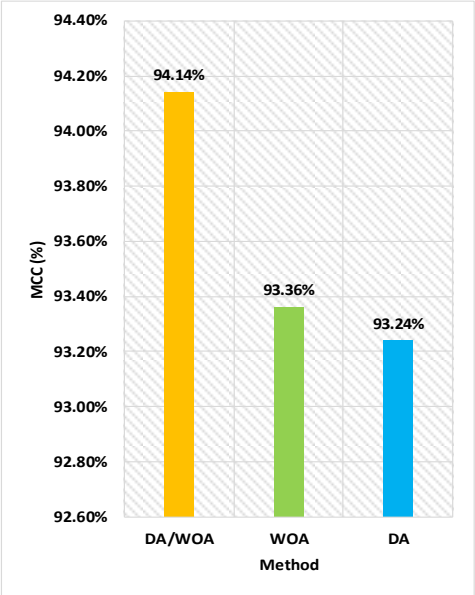

Fig. 8. MCC of the proposed phishing websites detection model.

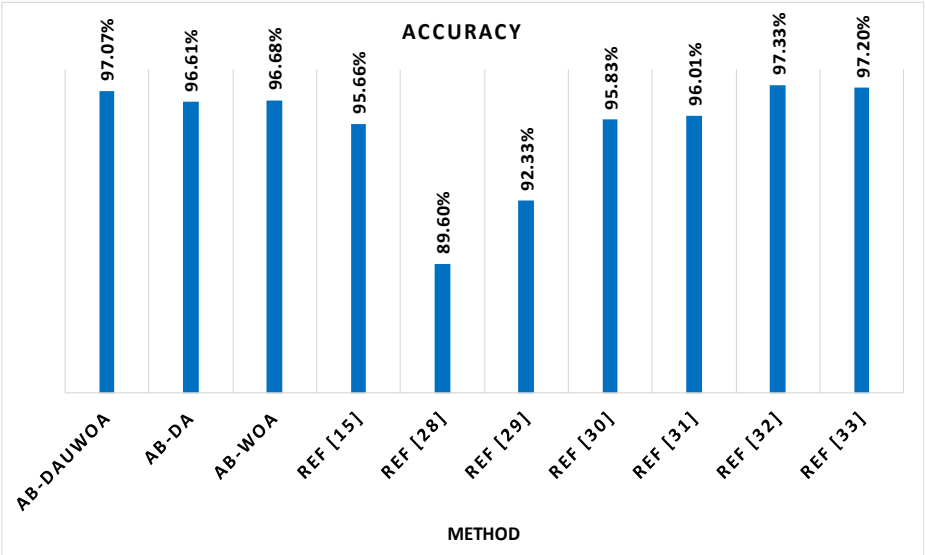

Fig. 9. Accuracy of the proposed AB-DAUWOA model against other phishing detection models.

VI. Runtime Efficiency

The results in Table VI show that the hybrid DAUWOA method delivers competitive runtime performance despite combining two optimization algorithms. Using the AB classifier as an example, the training time under DAUWOA was 0.897 s, which falls between DA (0.344 s) and WOA (0.914 s). The inference

time remained low at 0.005 s, very close to DA (0.004 s) and faster than WOA (0.008 s). Total execution time was also efficient (0.027 s), matching DA and outperforming WOA (0.036 s). These numbers indicate that the hybrid approach adds minimal overhead compared to running each algorithm alone.

TABLE VI: RUNTIME EFFICIENCY AB CLASSIFIER

Method	Training Time(s)	Inference Time(s)	Execution Time(s)
DA	0.344	0.004	0.027
WOA	0.914	0.008	0.036
DAUWOA	0.897	0.005	0.027

The DAUWOA model has a time complexity of $O(N \times F \times I)$, where N is the number of agents (population size), F is the number of features, and I is the number of iterations, with WOA and DA running independently before being merged through a union step. This keeps the computation linear with minimal overhead and allows easy parallelization, enabling the method to scale well to larger or high-dimensional datasets. As a result, DAUWOA offers an effective balance between runtime, convergence, and accuracy, making it suitable for fast, real-time phishing detection and similar cybersecurity tasks.

VII. LIMITATIONS AND FUTURE WORK

Although the proposed model shows promising results, several practical limitations remain. One challenge is the imbalance that often appears in real-world phishing datasets, where malicious samples are far fewer than legitimate ones. This imbalance can influence how the classifier behaves once deployed and may increase the chance of missing new attacks. Another limitation is the model's exposure to adversarial URLs in real-time settings. Attackers frequently adjust or disguise URL components in subtle ways, which may reduce the model's reliability if it encounters patterns that differ from those seen during training.

Looking ahead, this opens several directions for improvement. Future work should explore methods for handling skewed datasets more effectively, as well as strategies that strengthen the model against adversarial manipulation. It would also be valuable to test the approach on additional, previously unseen datasets to better understand how well it generalizes beyond the ISCX-URL2016 dataset. Finally, applying the same feature-selection framework to related cyber-threat areas—such as spam or malware detection—could help assess its broader usefulness and reveal where further refinements are needed.

VIII. CONCLUSION

This study introduced a robust phishing detection framework built around a hybrid feature-selection method (DAUWOA) and the AB classifier. By integrating the complementary strengths of DA and WOA, the proposed FS approach effectively reduced the original high-dimensional feature space while preserving the most discriminative attributes. This reduction directly addressed key challenges in phishing detection—such as model complexity, overfitting risks, and noisy URL representations—leading to improved learning stability and better generalization. Using the ISCX-URL2016 dataset, the DAUWOA-based model consistently outperformed individual methods and existing approaches across accuracy, recall, precision, F1-score, and MCC. The findings confirm that hybrid feature selection plays a critical role in simplifying the learning task and enhancing detection performance, making the proposed method a strong candidate for deployment in practical, real-time phishing-detection environments.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All authors contribute evenly to this paper. All authors had approved the final version.

REFERENCES

[1] H. N. Mohammed, N. S. Malami, S. Thomas, F. A. Aiyelabegan, F. A. Imam and H. H. Ginsau, "Machine learning approach to anti-money laundering: A review," in *Proc. 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development*, Lagos, Nigeria, 2022. doi: 10.1109/NIGERCON54645.2022.9803072

[2] A. H. Al-Qassem, M. H. Ryad, Z. Alkhazali *et al.*, "The impact of technological advancements on human resource management practices: Adapting to the digital era," *Data & Metadata*, vol. 4, 731, 2025.

[3] Statista. (May 2023). Digital Payments: market data & analysis. [Online]. Available: https://www.statista.com/outlook/fmo/digital-payments/worldwide/

[4] T. J. Ayeni, E. O. Durotoye, and S. Eriabie, "Adoption of artificial intelligence for fraud detection in deposit money banks in Nigeria," in *Proc. 2024 Int. Conf. on Science, Engineering and Business for Driving Sustainable Development Goals*, Omu-Aran, Nigeria, 2024. doi: 10.1109/SEB4SDG60871.2024.10630329

[5] Y. A. Maz, M. Anbar, S. Manickam *et al.*, "Transfer learning-based approach with an ensemble classifier for detecting keylogging attack on the internet of things," *Computers, Materials & Continua*, vol. 85, no. 3, pp. 5287–5307 2025.

[6] Petrosyan. (Feb. 3, 2025). Estimated annual cost of cybercrime in the United States from 2017 to 2028. *Statista*. [Online]. Available: https://www.statista.com/forecasts/1399040/us-cybercrime-cost-annual

[7] D. Rathee and S. Mann, "Detection of E-mail phishing attacks—using machine learning and deep learning," *International Journal of Computer Applications*, vol. 183, no. 47, pp. 1–7, Jan. 2022.

[8] M. M. Abualhaj, S. N. Al-Khatib, A. A. Abu-Shareha *et al.*, "Spam detection boosted by firefly-based feature selection and optimized," *Int. J. Adv. Soft Comput. Appl*., vol. 17, no. 3, pp. 1–19, Nov. 2025.

[9] Petrosyan. (Dec. 9, 2024). Number of unique phishing sites detected worldwide from 3rd quarter 2013 to 3rd quarter 2024. *Statista*. [Online]. Available: https://www.statista.com/statistics/266155/ number-of-phishing-domain-names-worldwide/

[10] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha and M. Guizani, "Systematization of Knowledge (SoK): A systematic review of software-based web phishing detection," *EEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2797–2819, 2017.

[11] R. W. Purwanto, A. Pal, A. Blair and S. Jha, "PhishSim: Aiding phishing website detection with a feature-free tool," *IEEE Trans. on Information Forensics and Security*, vol. 17, pp. 1497–1512, 2022. doi: 10.1109/TIFS.2022.3164212

[12] M. M. Abualhaj, S. N. Al-Khatib, A. A. Abu-Shareha, A. Hyassat, and M. Sh. Daoud, "Smart firewall for phishing detection powered by bio-inspired algorithms," *Journal of Advances in Information Technology*, vol. 16, no. 11, pp. 1529–1539, 2025.

[13] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics," *Expert Systems with Applications*, vol. 236, 121183, Feb. 2024.

[14] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," arXiv preprint, arXiv:2009.11116, 2020.

[15] J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing detection using machine learning technique," in *Proc. 2020 First International Conference of Smart Systems and Emerging Technologies*, Riyadh, Saudi Arabia, 2020, pp. 43–46. doi: 10.1109/SMART-TECH49988.2020.00026

[16] R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Enhanced multiclass android malware detection using a modified dwarf mongoose algorithm", *International Journal of Analysis and Applications*, vol. 23, pp. 1–23, Jan. 2025.

[17] R. Zieni, L. Massari and M. C. Calzarossa, "Phishing or not phishing? A survey on the detection of phishing websites," *IEEE Access*, vol. 11, pp. 18499–18519, 2023.

[18] Y. Wang, M. A. Elmeligy, H. F. Isleem *et al.*, "Modeling of concrete-filled PVC tube columns confined with CFRP strips under uniaxial eccentric compression: Machine learning and finite element approaches," *Journal of Big Data*, vol. 12, no. 1, 34, 2025.

[19] Y. Sanjalawe, S. Fraihat, S. Al-E'mari *et al.*, "Smart load balancing in cloud computing: Integrating feature selection with advanced deep learning models," *PLOS One*, vol. 20, no. 9, e0329765, Sep. 2025.

[20] A. Zraiqat, O. Sayyed, M. Soudi *et al*., "Key maker algorithm: A novel human-based metaheuristic for constrained optimization," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 9, pp. 688–699, 2025.

[21] A. Zraiqat, B. Batiha, O. Al-Refai *et al.*, "Psychologist algorithm: A human-inspired metaheuristic for solving complex constrained optimization problems," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 9, 2025. doi: 10.22266/ijies2025.1031.09

[22] A. Zraiqat, O. Sayyed, M. Soudi *et al*., "Driver and navigator algorithm: A novel parameter-free human-inspired metaheuristic for efficient global optimization," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 9, pp. 555–569, 2025.

[23] M. Arabasy and R. S. Ghoneim, "Enhancing sustainable urban design using machine learning: Comparative analysis of seven metaheuristic algorithms in energy-efficient digital architecture," *Frontiers in Built Environment*, vol. 11, 1526209, 2025.

[24] M. M. Abualhaj, S. N. Al-Khatib, M. Al-Zyoud *et al.*, "Enhanced network communication security through hybrid dragonfly–bat feature selection for intrusion detection," *Journal of Communications*, vol. 20, no. 5, pp. 607–618, 2025.

[25] O. Almomani, A. Alsaaidah, A. A. Abu-Shareha *et al.*, "Enhance URL defacement attack detection using particle swarm optimization and machine learning," *Journal of Computational and Cognitive Engineering*, Feb. 2025. doi: 10.47852/bonviewjcce52024668

[26] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious URLS using lexical analysis," in *Proc. 10th International Conference*, Taipei, Taiwan, 2016. doi: 10.1007/978-3-319-46298-1_30

[27] M. M. Abualhaj, A. A. Abu-Shareha, S. N. Al-Khatib, Q. Y. Shambour, and A. M. Alsaaidah, "Detecting spam using Harris Hawks optimizer as a feature selection algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 2361–2369, May 2025.

[28] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, Sep. 2018. doi: https://doi.org/10.1155/2018/4678746

[29] H. Alqahtani and A. Abu-Khadrah, "Enhance the accuracy of malicious uniform resource locator detection based on effective machine learning approach," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 6, pp. 4422–4429, Sep. 2024.

[30] M. A. Daniel, S.-C. Chong, L.-Y. Chong, and K.-K. Wee, "Optimising phishing detection: A comparative analysis of machine learning methods with feature selection," *Journal of Informatics and Web Engineering*, vol. 4, no. 1, pp. 200–212, Feb. 2025.

[31] N. Nagy, M. Aljabri, A. Shaahid *et al.*, "Phishing URLs detection using sequential and parallel ML techniques: Comparative analysis," *Sensors*, vol. 23, no. 7, 3467, 2023.

[32] M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Information Security*, vol. 17, no. 3, pp. 423–440, 2023.

[33] R. Vinayakumar, S. Srinivasan, K. P. Soman, and M. Alazab, "Malicious URL detection using deep learning," TechRxiv, preprint, 2023. doi: 10.36227/techrxiv.19308183.v1

[34] M. Abualhaj, M. Al-Zyoud, M. Hiari *et al.*, "A fine-tuning of decision tree classifier for ransomware detection based on memory data," *International Journal of Data and Network Science*, vol. 8, no. 2, pp. 733–742, 2024.

[35] A. Abu-Khadrah, M. A. AlMutairi, M. R. Hassan, and A. M. Ali, "Enhancing IoT security and malware detection based on machine learning," in *Proc. Int. Congress on Information and Communication Technology*, 2025, pp. 561–571.

[36] H. Chantar, M. Tubishat, M. Essgaer, and S. Mirjalili, "Hybrid binary dragonfly algorithm with simulated annealing for feature selection," *SN Computer Science*, vol. 2, no. 4, May 2021. doi: https://doi.org/10.1007/s42979-021-00687-5

[37] P. Nandal, P. Mann, N. Bohra, K. Sagar, and A. Smerat, "Improving underwater image quality through Real-ESRGAN with Whale optimization algorithm," *Internet Technology Letters*, vol. 8, no. 4, e70047, 2025.

[38] M. M. Abualhaj, S. N. Al-Khatib, M. Zyoud, I. Qaddara, and M. Anbar, "Enhancing intrusion detection system performance using a hybrid of Harris Hawks and Whale optimization algorithms," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 24354–24361, 2025.

[39] A. Arabiat and M. Altayeb, "Driving behavior analytics: An intelligent system based on machine learning and data mining techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 2055–2065, 2025.

[40] M. M. Al-Momani, T. A. Alqudah, I. A. Swiety *et al*., "Integrating Artificial Intelligence (AI) and Business Intelligence (BI): A framework for improving enterprise performance," *TEM Journal*, vol. 14, no. 3, pp. 2208–2216, 2025.

[41] P. Sornsuwit and S. Jaiyen, "A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting," *Applied Artificial Intelligence*, vol. 33, no. 5, pp. 462–482, Mar. 2019.

[42] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 1–12, Sep. 2021.

[43] M. Açikkar, "Fast grid search: A grid search-inspired algorithm for optimizing hyperparameters of support vector regression," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 32, no. 1, pp. 68–92, Feb. 2024.

[44] Y. Sanjalawe, S. Al-E'mari, S. Fraihat, and E. Alzubi, "A deep learning-driven multi-layered steganographic approach for enhanced data security," *Scientific Reports*, vol. 15, no. 1, Feb. 2025. doi: https://doi.org/10.1038/s41598-025-89189-5

[45] R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Android malware detection using a modified dwarf mongoose algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 8, 2025. doi: 10.22266/ijies2025.0930.21

[46] R. Tarek, A. Elshenawy, M. I. Assadwy, and M. A. Madkour, "Automated diagnosis of dental diseases using deep learning on radiographic images," *SN Computer Science*, vol. 6, no. 6, 751, 2025.

[47] R. Masadeh, O. Almomani, A. Zaqebah *et al.*, "Narwhal Optimizer: A nature-inspired optimization algorithm for solving complex optimization problems," *Computers, Materials & Continua*, vol. 85, no. 2, pp. 3709–3737, 2025.

[48] F. O. Al-znamat, A. Alsaaidah, and A. A. Abu-Shareha, "Malware detection using a modified bat algorithm for feature selection," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 9, pp. 885–899, 2025.

Mosleh M. Abu-Alhaj is a senior lecturer in Al-Ahliyya Amman University. He received his first degree in computer science from Philadelphia University, Jordan, in 2004, master degree in computer information system from the Arab Academy for Banking and Financial Sciences, Jordan in 2007, and Ph.D. degree in multimedia networks protocols from Universiti Sains Malaysia in 2011. His research area of interest includes VoIP, congestion control, and cybersecurity data mining and optimization.

Mohammad O. Hiari is a lecturer in Al-Ahliyya Amman University. He received his first degree in software engineering from Philadelphia University, Jordan, in August 2004 and master degree in computer science from Al Balqa Applied University, Jordan in February 2016. His research area of interest includes VoIP and cybersecurity data mining and optimization.

Sumaya Nabil Alkhatib is a senior lecturer in Al-Ahliyya Amman University. She received his first degree in computer science from Baghdad University, Iraq, in June 1994 and master degree in computer information system from the Arab Academy for Banking and Financial Sciences, Jordan in February 2007. Her research area of interest includes VoIP, congestion control, and cybersecurity data mining and optimization.

Ahmad Adel Abu-Shareha received his first degree in computer science from Al Al-Bayt University, Jordan, 2004, master degree from Universiti Sains Malaysia (USM), Malaysia, 2006, and Ph.D degree from USM, Malaysia, 2012. His research focuses on data mining, artificial intelligent and multimedia security. He investigated many machine learning algorithms and employed artificial intelligent in variety of fields, such as network, medical information process, knowledge construction and extraction.

Mohammad Sh. Daoud received the Ph.D. degree in computer science from De Montfort University, U.K. He is currently an associate professor with the College of Engineering, Al Ain University, United Arab Emirates. His research interests include artificial intelligence, swarm systems, secured systems and networks, and smart applications.

Muhammad Rehan Faheem received the M.S. degree in computer science in 2017 and the Ph.D. degree in 2024. He is currently working as a senior lecturer with the Faculty of Artificial Intelligence and Cybersecurity, Universiti Teknikal Malaysia Melaka. His research interests include machine learning, natural language processing, data security and intelligent systems. He has authored and coauthored several papers in well reputed international journals.

Mohammed Anbar received the B.Sc. degree in software engineering from Al-Azhar University, Palestine, in 2008, the M.Sc. degree in information technology from Universiti Utara Malaysia, in 2009, and the Ph.D. degree in advanced internet security and monitoring from Universiti Sains Malaysia (USM), in 2013. He is currently a senior lecturer with the National Advanced IPv6 Centre (NAv6), USM. His current research interests include malware detection, Intrusion Detection Systems (IDSs), Intrusion Prevention Systems (IPSs), network monitoring, the Internet of Things (IoT), Software-Defined Networking (SDN) security, cloud computing security, and IPv6 security.

Ali Al-Allawee is an associate professor in computer science. He earned his Ph.D. degree (2014) from University Science Malaysia and his master's degree from University Technology Malaysia. He further pursued postdoctoral research at the University of Haute-Alsace, France. His research focuses on networking, multimedia communication, cloud computing, and edge computing.