Smart Firewall for Phishing Detection Powered by Bio-Inspired Algorithms

Mosleh M. Abualhaj [1,*], Sumaya N. Al-Khatib [2], Ahmad A. Abu-Shareha [3], Abdallah Hyassat [3], and Mohammad Sh. Daoud [4]

[1] Department of Networks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[2] Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3] Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[4] College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
Email: m.abualhaj@ammanu.edu.jo (M.M.A.); sumayakh@ammanu.edu.jo (S.N.A.); a.abushareha@mmanu.edu.jo (A.A.A.); a.hyassat@ammanu.edu.jo (A.H.); mohammad.daoud@aau.ac.ae (M.S.D)
*Corresponding author

***Abstract*—Phishing attacks continue to pose significant risks to digital security by exploiting user vulnerabilities through deceptive methods. This paper presents a smart firewall model for phishing detection that leverages bio-inspired algorithms to enhance threat identification and response. The model utilizes the Whale Optimization Algorithm (WOA) and Dragonfly Algorithm (DA) independently for effective feature selection, thereby reducing data dimensionality while retaining critical phishing indicators. These optimized features are then processed by advanced Machine Learning (ML) classifiers—Extra Trees (ET), Random Forest (RF), and K-Nearest Neighbors (KNN)—to rigorously evaluate detection accuracy. Experimental results on the ISCX-URL2016 dataset demonstrate that the combination of WOA with the ETs classifier achieves a superior detection accuracy of 98.86%, precision of 99.50%, recall of 99.50%, F1-Score of 99.50%, outperforming alternative configurations and recent methods. This result highlights the potential of bio-inspired optimization combined with ML to develop intelligent, adaptive firewalls capable of effectively mitigating phishing threats.**

***Keywords*—smart firewall, phishing detection, bio-inspired algorithms, feature selection, machine learning**

I. Introduction

The digital realm has entered all facets of human life. People use digital tools in their communication, entertainment, education, and daily tasks. Nevertheless, intruders continually exploit the digital realm's vulnerabilities to cause harm. These intruders often adapt their tools and methodologies based on the weaknesses they discover. The damage caused by such intrusions is frequently destructive to digital services and infrastructure [1, 2]. For example, businesses lost approximately $8 trillion in 2022 due to cyber intrusions [3]. Phishing is one of the most sophisticated attacks used to infiltrate the digital environment. It deceives users into revealing sensitive personal information and is commonly delivered via email or fake websites [4]. According to recent statistics from 2025, each phishing breach incurs an average cost of approximately $5 million [5]. Consequently, security specialists must make significant efforts to reduce the impact and cost of phishing-based intrusions in cyberspace. Recently, tools and systems used to mitigate intrusions—including firewalls, system vulnerability analyzers, and endpoint security solutions—have been enhanced using advanced techniques [6–8].

A firewall is a vital component in most network security systems, designed to protect digital assets and IT infrastructure from intrusions. Traditional firewalls function by inspecting incoming network traffic and comparing specific fields (e.g., IP address and port number) in the packet header against predefined rules. These fields are examined and compared in sequence to the firewall rules. Then, packets are allowed or disallowed to enter the network depending on the corresponding rule [9, 10]. Nevertheless, the tools and techniques currently available to attackers enable them to bypass traditional firewalls with relative ease.

Hence, new mechanisms capable of detecting and preventing sophisticated intrusions should be incorporated into firewalls. One such approach is the integration of Artificial Intelligence (AI), particularly Machine Learning (ML), which can be embedded within firewall systems to analyze data and detect emerging patterns of phishing-based intrusions [7, 11, 12].

ML methods inspect and evaluate previously observed phishing and non-phishing scenarios to hinder future phishing-based intrusions. However, the complexity and large volume of data in phishing and non-phishing scenarios make the inspection and evaluation process challenging for ML methods. Also, a significant portion of

Manuscript received July 31, 2025; revised August 12, 2025; accepted August 21, 2025; published November 7, 2025.

the attributes of the phishing and non-phishing data may be redundant or irrelevant to intrusion detection [13, 14]. Accordingly, the performance of ML methods in detecting phishing-based intrusions can be negatively impacted. Therefore, the amount of phishing and non-phishing data should be optimized by retaining only the most relevant features that enhance the performance of the smart firewall (i.e., a firewall that uses AI). These features can be identified using ML feature selection mechanisms. In recent years, nature-inspired metaheuristic algorithms have been successfully employed as feature selection mechanisms [15, 16].

In recent years, many dilemmas in various fields have been solved using metaheuristic algorithms. In the field of cybersecurity, researchers have successfully used these algorithms to protect the digital realm. These methods have been designed based on several natural phenomena, including how animals acquire their food. The Whale Optimization Algorithm (WOA) and Dragonfly Algorithm (DA) are two animal-based metaheuristic algorithms that are consistently used in many cybersecurity solutions to protect the elements of the digital realm [8, 17, 18]. This article utilizes the WOA and DA algorithms to enhance the functionality of the proposed smart firewall. These two methods have been used for feature selection from phishing-related data to improve the accuracy of threat detection and prevention. In addition, Extra Trees (ET), K-Nearest Neighbors (KNN), and Random Forest (RF) classifiers have been utilized with the suggested smart firewall. The main contributions of this work are summarized as follows:

- A smart firewall–based phishing detection framework is proposed, integrating metaheuristic feature selection using the WOA and DA algorithms with three ML classifiers: ET, RF, and KNN.
- A direct comparative evaluation of WOA and DA feature selection methods is conducted using the ISCX-URL2016 dataset, providing insights into their relative effectiveness for phishing URL detection.
- Random Search hyperparameter tuning is applied to optimize classifier settings, ensuring that performance results reflect well-calibrated models.
- An extensive performance evaluation is performed, reporting accuracy, precision, recall, and F1-Score, enabling both predictive quality and computational efficiency assessments.

II. Background

This section outlines the ISCX-URL2016 phishing dataset, the WOA and DA feature selection algorithms, and the KNN, RF, and ET classifiers used in the proposed smart firewall.

A. ISCX-URL2016 Phishing Dataset

The ISCX-URL2016 dataset comprises phishing and legitimate URLs for detection research purposes. It includes 15,367 URL samples divided into 7586 phishing samples and 7781 benign samples. The dataset comprises 79 distinct features that describe the attributes of each URL, as outlined in Table I. These features effectively reflect both structural and lexical aspects of URLs, including domain-specific tokens that help identify suspicious URL components. Additionally, the dataset provides features that reflect URL obfuscation methods used by attackers. Some features require normalization before training models due to scale differences. While noise is minimal, manual cleaning is recommended for optimal results [14, 19, 20].

TABLE I. ISCX-URL2016 Phishing Dataset

#	Feature Name	Improved Description
1	Querylength	Length of the query string (after '?')
2	domain_token_count	Count of tokens in the domain (split by '.' or '-')
3	path_token_count	Count of tokens in the path (split by '/')
4	avgdomaintokenlen	Average length of domain tokens
5	longdomaintokenlen	Length of the longest domain token
6	avgpathtokenlen	Average length of path tokens
7	tld	Number of top-level domain segments (e.g., .com, .co.uk)
8	charcompvowels	Ratio of vowels to total characters in the URL
9	charcompace	Ratio of space-encoded characters (e.g., '%20') in the URL
10	ldl_url	Frequency of letter-digit-letter patterns in the URL
11	ldl_domain	Frequency of the same pattern in the domain
12	ldl_path	Frequency of the same pattern in the path
13	ldl_filename	Frequency of the same pattern in the file name
14	ldl_getArg	Frequency of the same pattern in query argument values
15	dld_url	Frequency of digit-letter-digit patterns in the URL
16	dld_domain	Frequency of the same pattern in the domain
17	dld_path	Frequency of the same pattern in the path
18	dld_filename	Frequency of the same pattern in the file name
19	dld_getArg	Frequency of the same pattern in argument values
20	urlLen	Total length of the full URL
21	domainlength	Total length of the domain
22	pathLength	Length of the path section
23	subDirLen	Length of sub-directory path (excluding file name)
24	fileNameLen	Length of the file name
25	this.fileExtLen	Length of the file extension (e.g., .html, .php)
26	ArgLen	Length of the query string (arguments)

27	pathurlRatio	Ratio of path length to full URL length
28	ArgUrlRatio	Ratio of argument length to URL length
29	argDomanRatio	Ratio of argument length to domain length
30	domainUrlRatio	Ratio of domain length to URL length
31	pathDomainRatio	Ratio of path length to domain length
32	argPathRatio	Ratio of argument length to path length
33	executable	Flag indicating if the URL points to an executable file
34	isPortEighty	Flag indicating if port 80 is explicitly mentioned
35	NumberofDotsinURL	Total number of dots (‘.’) in the URL
36	ISIpAddressInDomainName	Flag indicating if the domain is an IP address
37	CharacterContinuityRate	Ratio of longest run of same character type to URL length
38	LongestVariableValue	Length of the longest value in query parameters
39	URL_DigitCount	Number of digits in the full URL
40	host_DigitCount	Number of digits in the domain
41	Directory_DigitCount	Number of digits in the path
42	File_name_DigitCount	Number of digits in the file name
43	Extension_DigitCount	Number of digits in the extension
44	Query_DigitCount	Number of digits in the query string
45	URL_Letter_Count	Number of letters in the full URL
46	host_letter_count	Number of letters in the domain
47	Directory_LetterCount	Number of letters in the directory path
48	Filename_LetterCount	Number of letters in the file name
49	Extension_LetterCount	Number of letters in the extension
50	Query_LetterCount	Number of letters in the query string
51	LongestPathTokenLength	Length of the longest token in the path
52	Domain_LongestWordLength	Length of the longest token in the domain
53	Path_LongestWordLength	Length of the longest token in the path
54	sub-Directory_LongestWordLength	Length of the longest token in the sub-directory
55	Arguments_LongestWordLength	Length of the longest token in argument values
56	URL_sensitiveWord	Count of sensitive keywords (e.g., ‘login’, ‘bank’) in URL
57	URLQueries_variable	Number of key = value query parameters
58	spcharUrl	Count of special characters (e.g., @, %, &) in URL
59	delimeter_Domain	Count of delimiters in domain (e.g., ‘.’, ‘-’)
60	delimeter_path	Count of delimiters in path (e.g., ‘/’, ‘-’)
61	delimeter_Count	Total number of delimiters in the URL
62	NumberRate_URL	Ratio of digits to total length in the full URL
63	NumberRate_Domain	Ratio of digits to length in the domain
64	NumberRate_DirectoryName	Ratio of digits to length in the directory name
65	NumberRate_FileName	Ratio of digits to length in the file name
66	NumberRate_Extension	Ratio of digits to length in the extension
67	NumberRate_AfterPath	Ratio of digits to length in everything after the path
68	SymbolCount_URL	Count of symbols in the full URL
69	SymbolCount_Domain	Count of symbols in the domain
70	SymbolCount_Directoryname	Count of symbols in the directory name
71	SymbolCount_FileName	Count of symbols in the file name
72	SymbolCount_Extension	Count of symbols in the extension
73	SymbolCount_Afterpath	Count of symbols after the path
74	Entropy_URL	Shannon entropy of the full URL
75	Entropy_Domain	Shannon entropy of the domain
76	Entropy_DirectoryName	Shannon entropy of the directory name
77	Entropy_Filename	Shannon entropy of the file name
78	Entropy_Extension	Shannon entropy of the extension
79	Entropy_Afterpath	Shannon entropy of everything after the path

The dataset is publicly available and widely cited in phishing detection research. It is suitable for supervised learning classification tasks such as phishing detection. The dataset size supports robust ML model training and evaluation, and it is well-balanced with an adequate number of samples in each target class. The high number of attributes enables feature selection studies, and the dataset is ideal for testing generalization in URL-based phishing classifiers [14, 19, 20].

B. Feature Selection with DA and WOA Algorithms

WOA is a nature-inspired metaheuristic based on humpback whale behavior. It simulates a bubble-net feeding strategy for search and optimization. WOA balances exploration and exploitation using adaptive spiral and encircling mechanisms. Global search is modeled via random search agents moving away from prey, while local search mimics shrinking, encircling, and spiral-shaped prey attacks. Additionally, WOA requires minimal control parameters: population size and maximum iterations [8, 21]. On the other hand, DA is a swarm intelligence algorithm inspired by the static and dynamic swarming behaviors of dragonflies. It simulates five main behaviors: separation, alignment, cohesion, attraction to food, and distraction from enemies. Exploration is modeled by dynamic (global) swarming to search the solution space, whereas exploitation is achieved through static (local) swarming around promising solutions. Position updates utilize Lévy flight and velocity vectors to achieve a balance between convergence and exploration. WOA and DA have been widely applied in feature selection, classification, scheduling, and clustering. Table II compares and contrasts the WOA and DA in feature selection [8, 17, 21, 22].

TABLE II. COMPARATIVE ANALYSIS OF WOA AND DA FOR FEATURE SELECTION TASKS

Aspect	WOA	DA	Relevance to Feature Selection
Algorithm Type	Nature-inspired, population-based	Swarm intelligence-based	Determines the method's ability to search high-dimensional feature spaces effectively.
Main Operators	Encircling prey, spiral update, random search	Separation, alignment, cohesion, attraction, distraction	Operators dictate how features are explored, selected, and refined over iterations.
Exploration Strategy	Agents randomly diverge from prey (global optima)	Dynamic swarming with Lévy flights	Controls the ability to escape local optima and explore diverse feature subsets.
Exploitation Strategy	Spiral attack and shrinking encircling near prey	Static swarming around promising solutions	Improves selection precision by focusing on the most promising features.
Position Update Formula	Combines distance and spiral equations	Vector-based with behavioral weights and velocity	Defines how candidate feature subsets evolve; impacts convergence accuracy and speed.
Population Interaction	Indirect (distance to best solution)	Direct (neighbor interactions)	Influences diversity of solutions, which is crucial in avoiding redundancy in selected features.
Information Sharing	Only through best solution (leader)	Through neighboring individuals	More interaction improves solution diversity, aiding robust feature selection.
Convergence Behavior	Fast but may converge early	More stable, slower convergence	Stability ensures reliable selection of relevant features; avoids overfitting.

C. ML Classifier Used in the Proposed Smart Firewall

KNN, RF, and ET are employed in the proposed smart firewall. The basic idea behind KNN is to classify new data points based on the majority class of the K closest data points in the training set. Fig. 1 illustrates the KNN classifier. The choice of K, or the number of neighbors to consider, is a hyperparameter that can be tuned for optimal performance. For each new data point, KNN measures its distance to all training samples using metrics such as Euclidean, Manhattan, or Minkowski. Euclidean distance (Eq. (1)) is the default metric commonly used in KNN implementations [23, 24].

$$\text{Euclidean Distance between X and Y} = \sqrt{(A_2 - A_1)^2 + (B_2 - B_1)^2} \quad (1)$$

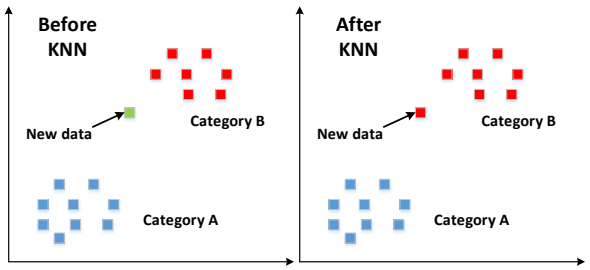

Fig. 1. Conceptual diagram of KNN algorithm.

RF is an ML ensemble algorithm that is used for regression and classification applications. As seen in Fig. 2, it is built on the idea of Decision Trees (DTs) and combines various DTs to increase the model's accuracy and robustness. The fundamental principle of RF is to generate a large number of DTs from the training dataset using randomly chosen characteristics. The final prediction is obtained by averaging the predictions of all the trees in the forest after each DT has been trained on a random portion of the training data and features. This procedure helps reduce overfitting and enhances the model's ability to generalize [25, 26].

ET is an ensemble algorithm built on DT foundations, similar to RF, and constructs many unpruned DTs for both classification and regression tasks. While ET and RF both rely on multiple randomized DTs, ET introduces greater randomness by utilizing the full training dataset rather than bootstrapped samples and selecting split thresholds randomly instead of searching for the best ones. This approach allows ET to train faster and often makes it more computationally efficient than RF. However, this added randomness can lead to slightly higher bias and increased instability compared to RF, which is generally more stable. Despite this, ET's randomness can improve generalization on noisy datasets and often yields better performance when features are highly correlated. In both algorithms, Gini impurity (Eq. (2)) is commonly used to evaluate the quality of splits, where G is Gini impurity, C is the number of classes, and pi is the proportion of samples belonging to class i [27, 28].

$$G = 1 - \sum_{i=1}^{c} p_i^2 \quad (2)$$

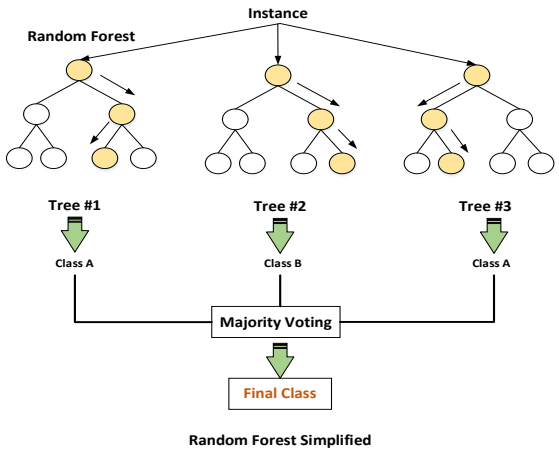

Fig. 2. Conceptual diagram of RF algorithm.

Overall, KNN is simple and effective for small datasets but slow for large ones, while RF is a robust ensemble method that reduces overfitting by using many trees and scales better with increasing data. ET offers faster training with more randomness and handles noisy, large data well, but can be less stable than RF. Table III shows the key hyperparameters and their values for the ET, RF, and KNN classifiers.

TABLE III. HYPERPARAMETER SETTINGS FOR ET, RF, AND KNN CLASSIFIERS

Hyperparameter	ET	RF	KNN
n_estimators	100	100	N/A
max_features	'sqrt'	'sqrt'	N/A
max_depth	None	None	N/A
min_samples_split	2	2	N/A
min_samples_leaf	1	1	N/A
bootstrap	False	True	N/A
criterion	'gini'	'gini'	N/A
n_neighbors	N/A	N/A	5
weights	N/A	N/A	'uniform'
metric	N/A	N/A	'euclidean'

III. RELATED WORKS

Several approaches have been proposed to handle phishing-based intrusions. The studies by Prabakaran *et al.* [29], Bu and Kim [30], and Uçar *et al.* [31] were evaluated using the same dataset as this study, namely the ISCX-URL2016 phishing dataset. Prabakaran *et al.* [29] have proposed a hybrid Deep Learning (DL) model combining a Variational Autoencoder (VAE) and a Deep Neural Network (DNN). VAE learns and extracts compact latent features from URLs, while DNN classifies URLs using VAE-extracted features. The hybrid DL model achieved 97.45% accuracy, outperforming other DL baselines. These DL baselines include Convolutional Neural Network (CNN) with 93.18% accuracy, Vanilla AE with 94.73%, Sparse AE with 95.09%, and Denoising AE with 95.71%. Bu and Kim [30] have proposed another hybrid model combining DL and Genetic Algorithms (GA). The GA selects the most relevant URL features, while the Convolutional–Recurrent Neural Network (CRNN) models learn and detect phishing patterns based on those optimized features. The hybrid CRNN DL-GA model achieved an accuracy of 96.85%. Uçar *et al.* [31] have proposed a DL model that automatically learns and extracts relevant features from the normalized URL data. The model compares the performance of CNN and Long Short-Term Memory (LSTM) models for phishing classification. CNN outperformed LSTM by achieving 93.67% accuracy, while LSTM achieved 88.25% accuracy.

The works proposed by Rashid *et al.* [32], Alqahtani and Abu-Khadrah [33], Zamir *et al.* [34], Daniel *et al.* [35], and Nagy *et al.* [36] have been evaluated using various phishing datasets and ML models. Rashid *et al.* [32] applied Principal Component Analysis (PCA) to a private dataset, reducing the original feature space to a lower-dimensional representation while preserving the most informative components. Subsequently, Support Vector Machine (SVM) and RF classifiers were used to identify phishing-based intrusions. The SVM classifier achieved the highest classification accuracy at 95.66%, outperforming the RF classifier, which achieved 93.22%. Alqahtani and Abu-Khadrah [33] proposed an ensemble model that integrates RF, SVM, and Bagging classifiers. First, the Pearson Correlation Coefficient (PCC) was used as a feature selection method to retain features that significantly contribute to phishing attack classification. Then, the ensemble model was tested on a public phishing dataset from the UCI Repository. The ensemble model achieved an accuracy of 92.33%. Zamir *et al.* [34] have proposed a new multi-method feature selection approach that combines information gain, gain ratio, Relief-F, and Recursive Feature Elimination (RFE). Then, PCA was applied to reduce the dimensionality of the resulting feature set. The study also proposed two stacking ensemble models for phishing detection. Stacking1 integrates RF, Neural Network (NN), and Bagging, while Stacking2 integrates RF, KNN, and Bagging. These stacking models employed a set of feature selection methods to identify the key features from a publicly available phishing dataset. Stacking1 achieved the highest classification accuracy of 97.4%, while Stacking2 followed closely with an accuracy of 97.2%. Daniel *et al.* [35] integrated PCA and RFE to identify and retain the most significant features for phishing detection based on a public Kaggle dataset. The RF classifier was then implemented on the reduced dataset, achieving an accuracy of 95.83%. Nagy *et al.* [36] proposed a model that employs a multi-feature selection approach combining Analysis of Variance (ANOVA), Chi-square, and correlation analysis. The model employs RF and Naïve Bayes (NB) classifiers to identify phishing-based intrusions. The RF classifier achieved 95.14% accuracy, while NB achieved 96.01% accuracy on the Malicious and Benign Webpages dataset.

The works proposed by Bu and Kim [30], Daniel *et al.* [35], Nagy *et al.* [36], and Yi [37] have been evaluated using various phishing datasets and DL models. Bu and Kim [30] evaluated the hybrid DL-GA model proposed using the PhishStorm and PhishTank datasets, achieving accuracies of 95.05% and 94.83%, respectively. Daniel *et al.* [35] evaluated the proposed method using Artificial Neural Network (ANN) classifiers, achieving an accuracy of 95.07%. Nagy *et al.* [36] evaluated the approach using CNN and LSTM, achieving accuracies of 95.13% and 95.14%, respectively. Yi *et al.* [37] have proposed a phishing detection model that adopts a dual-feature approach, integrating original URL-based features with interaction-based network features to enhance detection performance. The dual-feature approach was applied to a private dataset, and Deep Belief Networks (DBN) were utilized for classification, achieving an accuracy of 89.6%. Table IV summarizes the existing approaches for detecting phishing attacks.

Although prior studies have shown the effectiveness of various machine learning and deep learning models for phishing detection, several limitations remain. Many rely on static feature sets without comparing the performance of different metaheuristic feature selection algorithms. Few works have examined the relative strengths of distinct metaheuristics on the same dataset under consistent settings. This study addresses these gaps by directly comparing WOA and DA algorithms feature selection methods, evaluating their impact on multiple classifiers, optimizing hyperparameters, and assessing predictive performance using a standardized dataset.

TABLE IV. SUMMARY OF EXISTING APPROACHES FOR PHISHING ATTACK DETECTION

Dataset Context	Reference	Model	Feature Reduction Method	Dataset Name	Result (Accuracy) (%)
ISCX-URL2016 phishing dataset	Prabakaran *et al.* [29]	DNN	VAE	ISCX-URL2016 dataset	97.45
		CNN			93.18
		Vanilla AE			94.73
		Sparse AE			95.09
		Denoising AE			95.71
	Bu and Kim [30]	CRNN	GA algorithm	ISCX-URL2016	96.85
	Uçar *et al.* [31]	LSTM	Implicitly using LSTM	ISCX-URL2016	88.25
		CNN	Implicitly using CNN		93.67
Various phishing datasets and ML models	Rashid *et al.* [32]	SVM	PCA	Private dataset	95.66
		RF			93.22
	Alqahtani and Abu-Khadrah [33]	Ensemble	PCC	Public dataset	92.33
	Zamir *et al.* [34]	Stacking1	Multi-method selection method	Public dataset	97.4
		Stacking2			97.2
	Daniel *et al.* [35]	RF	PCA with RFE	Public dataset	95.83
	Nagy *et al.* [36]	NB	Multi-method selection method	Malicious and Benign Webpages dataset	96.01
		RF			95.14
Various phishing datasets and DL models	Bu and Kim [30]	CRNN	GA algorithm	PhishStorm	95.05
				PhishTank	94.83
	Daniel *et al.* [35]	ANN	PCA with REF	Public dataset	95.07
	Nagy *et al.* [36]	CNN	Multi-method selection method	Malicious and Benign Webpages dataset	95.13
		LSTM			95.14
	Yi *et al.* [37]	DBN	Implicitly using DBN	Private dataset	89.6

IV. THE PROPOSED SMART FIREWALL ML MODEL

The proposed phishing smart firewall ML detection model comprises three primary phases (As shown in Fig. 3): data preprocessing, feature selection, and classification, each designed to ensure high predictive performance and computational efficiency. This section provides a detailed explanation of the operations within each phase.

In the data preprocessing phase, the raw URL-phishing dataset, consisting of 79 features, undergoes two preparation steps to enhance its quality and ensure suitability for ML models. Initially, ten features are manually eliminated due to the presence of noisy or non-numeric values such as NaN, Infinity, or undefined tokens, which can adversely impact learning algorithms. The removed features include: 6) avgpathtokenlen, 32) argPathRatio, 64) NumberRate_DirectoryName, 65) NumberRate_FileName, 66) NumberRate_Extension, 67) NumberRate_AfterPath, 76) Entropy_DirectoryName, 77) Entropy_Filename, 78) Entropy_Extension, and 79) Entropy_Afterpath. Following feature elimination, normalization is performed using the min-max scaling technique, which rescales all feature values to a uniform range between 0 and 1. This normalization ensures proportional contribution of features to the model's learning process [23, 38]. For illustration, a sample feature vector before normalization contain the following values: {[80, 0.025, 5.083,333,5, 54, 48], [78, 0.025,641,026, 3.176,470,5, 54, 40], [71, 0.098,591,55, 6, 57, 22]}, and after normalization, it would be transformed to {[0.138,263,666, 0.027,443,609, 0.154,322,353, 0.154,121,864, 0.157,894,737], [0.131,832,797, 0.028,147,292, 0.096,350,222, 0.154,121,864, 0.131,578,947], [0.109,324,759, 0.108,228,319, 0.182,190,694, 0.164,874,552, 0.072,368,421]}.

At this stage, the dataset is normalized and comprises 69 features and is ready for the feature selection phase. In the feature selection phase, the DA and WOA metaheuristic optimization algorithms will be employed to identify the most informative features while eliminating those that are irrelevant or redundant. Each of these nature-inspired algorithms explores feature space to find subsets that maximize classification performance. The WOA and DA algorithms are applied independently to the full normalized dataset. As a result, WOA produces a reduced dataset with 43 selected features, represented by feature indices: {1, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 28, 29, 30, 33, 34, 35, 36, 37, 38, 40, 43, 45, 46, 47, 48, 49, 51, 52, 53, 55, 57, 58, 59, 62, 63, 69, 71, 72, 74, 75}. Similarly, DA identifies a slightly larger subset of 46 features, indexed as: {4, 7, 8, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 24, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 43, 44, 46, 47, 48, 52, 53, 54, 55, 56, 58, 62, 68, 69, 71, 72, 74, 75}. The application of both algorithms enables a comparative analysis and insights into their effectiveness on classification performance.

Finally, ET, RF, and KNN classifiers will be employed in the classification phase to evaluate the predictive capability of the feature-reduced datasets generated by WOA and DA algorithms. Each classifier is trained and assessed using 5-fold cross-validation. This robust resampling method partitions the dataset into five mutually exclusive subsets, ensuring that each instance is used for both training and validation exactly once. This technique provides a more reliable estimate of model performance [8, 23]. The classification results are assessed using four standard evaluation metrics: accuracy, recall, precision, and F1-Score. These metrics collectively offer a comprehensive evaluation of each model's effectiveness in phishing URL detection.

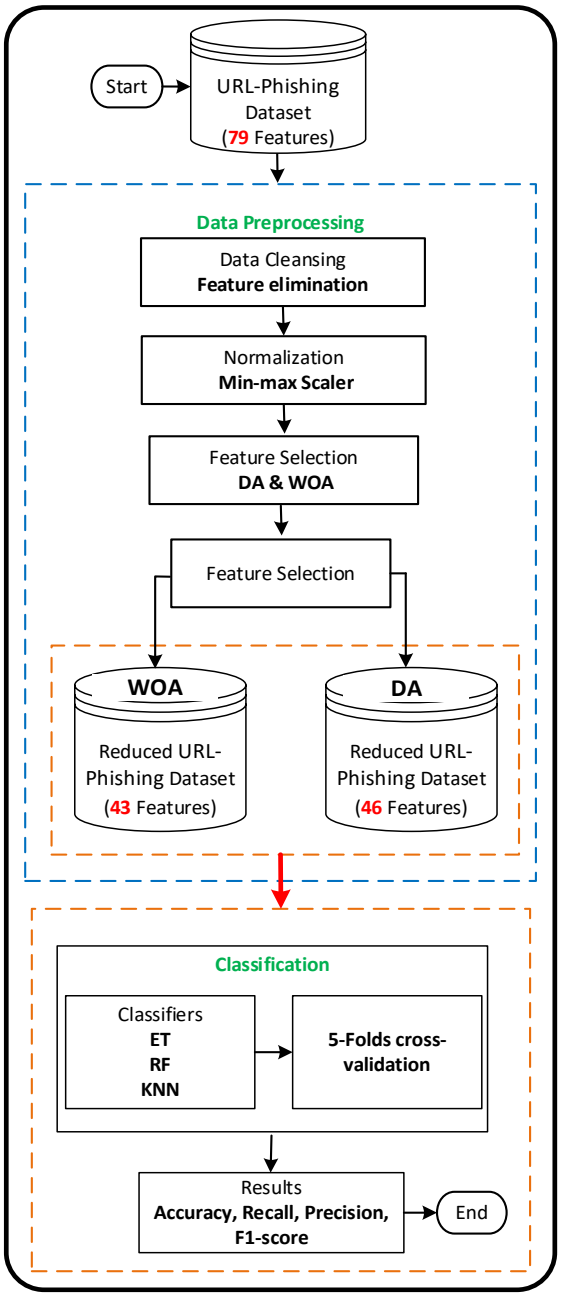

Fig. 3. Architecture of the proposed smart firewall–based phishing detection model.

V. Result and Discussion

The proficiency of the presented smart firewall model has been analyzed using Samsung Galaxy Book4 Ultra 16 AMOLED Touch Screen Laptop with Intel Core Ultra 9 Processor 185H (5.1 GHz speed, 16 cores, 22 threads, and 24 MB cache), 32GB DDR5 5600MT/s Memory, NVIDIA GeForce RTX 4070, and 1TB SSD. Ubuntu 20.04.6 LTS and Python 3.9.21 were used to implement the proposed smart firewall model. Ubuntu Linux was selected for its ability to efficiently handle computational workloads, automate repetitive tasks via shell scripts, support a wide range of ML models and libraries optimized for Linux, and enable efficient parallel processing for ML tasks. Conversely, the development of the proposed firewall model primarily relies on methods already available in Python packages such as Pandas, NumPy, Seaborn, TensorFlow, DEAP, and Scikit-learn.

A confusion matrix evaluates the performance of binary classification of the smart firewall model. It includes True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP and TN represent correct predictions, while FP and FN indicate errors. This helps assess model performance by visualizing correct versus incorrect classifications. Moreover, it forms the basis for calculating accuracy, precision, recall, and F1-Score. The evaluation metrics derived from the confusion matrix are shown in Table V [12, 39–41].

A. Evaluation of the Proposed Smart Firewall

In this subsection, the performance of the proposed smart firewall is assessed using a combination of the ET, RF, and KNN classifiers with the No Feature Selection (NoFS), DA, and WOA feature selection methods. Four evaluation metrics will be used to assess the performance of the proposed smart firewall, namely, accuracy, precision, recall, and F1-Score. These metrics provide insights into various aspects of the effectiveness of the proposed smart firewall.

Fig. 4 displays the accuracy of the proposed smart firewall model. The highest accuracy of 98.86% was achieved using the ET classifier with the WOA algorithm, which slightly outperformed ET with NoFS (98.76%) by 0.10% and ET with DA (98.50%) by 0.36%. Similarly, the RF classifier achieved its best result with WOA at 98.83%, which is 0.26% higher than RF with NoFS (98.57%) and 0.49% higher than RF with DA (98.34%). The KNN classifier showed overall lower performance across all methods, reaching its highest accuracy of 97.49% with WOA, which is 0.16% and 0.39% higher than its accuracy with DA (97.33%) and NoFS (97.10%), respectively.

TABLE V. Summary of Accuracy, Precision, Recall, and F1-Score Metrics

Metric	Description	Equation	When to Use	Interpretation
Accuracy	Overall correctness of the model.	(TP + TN)/(TP + TN + FP + FN)	When classes are balanced and all types of errors are equally important.	High value indicates most predictions are correct.
Precision	Correctness of positive predictions.	TP/(TP + FP)	When false positives are costly, e.g., in spam detection.	High precision means few irrelevant items are labeled as positive.
Recall	Ability to find all actual positives.	TP/(TP + FN)	When false negatives are costly, e.g., in disease or fraud detection.	High recall means most actual positives are correctly identified.
F1-Score	Harmonic mean of precision and recall.	2 × (Precision × Recall)/(Precision + Recall)	When a balance between precision and recall is needed.	High F1-Score indicates a good balance between precision and recall.

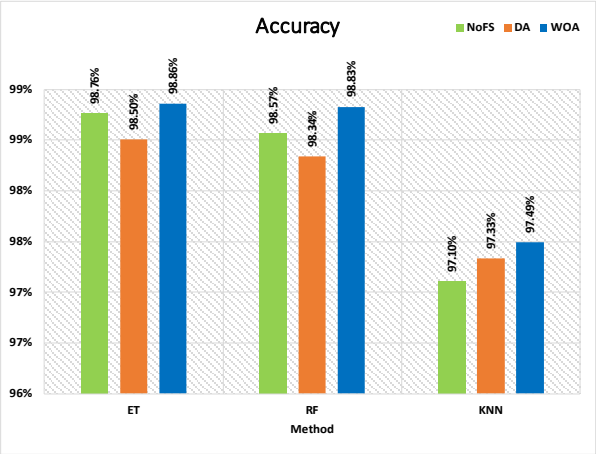

Fig. 4. Accuracy of the proposed smart firewall–based phishing detection model.

The high accuracy indicates that the model correctly classified a large majority of both phishing (TPs) and legitimate (TNs) URLs, while minimizing FPs—legitimate URLs incorrectly flagged as phishing—and FNs—phishing URLs missed by the firewall system. This balance is critical in phishing detection, as high False Positive Rates (FPRs) can disrupt user experience, while high False Negative Rates (FNRs) can leave firewall systems vulnerable to attacks. The strong accuracy results, especially with the ET classifier combined with WOA feature selection, reflect the proposed smart firewall's ability to detect threats effectively and reliably.

Fig. 5 displays the recall of the proposed smart firewall model. The highest recall of 98.61% was achieved using the RF classifier with the WOA algorithm, which slightly outperformed RF with NoFS (98.08%) by 0.53% and RF with DA (97.89%) by 0.72%. The ET classifier reached its highest recall with NoFS at 98.61%, which was 0.06% higher than ET with WOA (98.55%) and 0.53% higher than ET with DA (98.08%). The KNN classifier achieved the lowest recall across all methods, with its highest value at 96.50% using WOA, outperforming KNN with DA (96.10%) by 0.40% and with NoFS (95.97%) by 0.53%. The high recall indicates that the smart firewall correctly identified the majority of phishing attempts (TPs) while minimizing FNs—i.e., phishing URLs that went undetected. This capability is crucial in cybersecurity systems, as high recall ensures that actual threats are not missed. In the context of smart firewall phishing detection, this translates into more robust protection against attacks, reducing the risk of undetected breaches.

Fig. 6 displays the precision of the proposed smart firewall model. The highest precision of 99.14% was achieved using the ET classifier with the WOA algorithm, slightly outperforming ET with both NoFS and DA (98.87%) by 0.27%. The RF classifier achieved its highest precision of 99.00% with both NoFS and WOA, which was 0.27% higher than that of RF with DA (98.73%). The KNN classifier showed slightly lower precision across all methods, with the best result of 98.44% using DA, outperforming KNN with WOA (98.38%) by 0.06% and with NoFS (98.11%) by 0.33%. High precision indicates that when the smart firewall flags a URL as phishing, it is very likely to be correct. i.e., there are fewer FPs (legitimate URLs mistakenly identified as phishing). This minimizes disruption to users by reducing unnecessary blocking of safe websites. In the context of the smart firewall phishing detection system, high precision ensures that alerts are reliable and reduces the burden of reviewing false warnings.

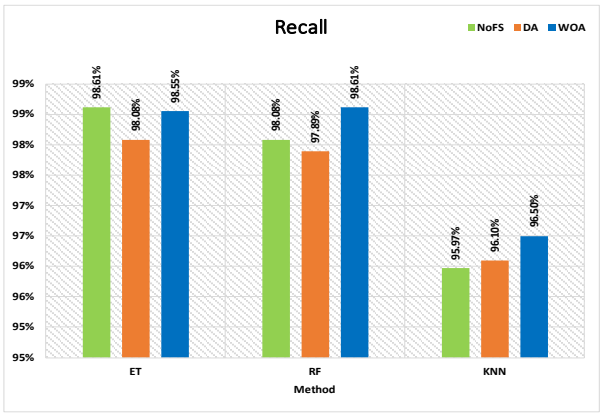

Fig. 5. Recall of the proposed smart firewall–based phishing detection model.

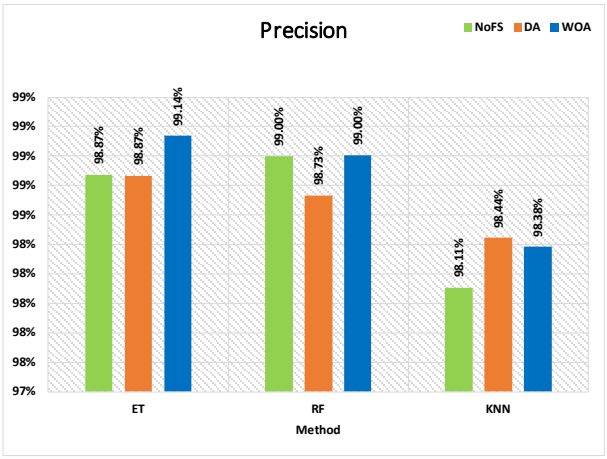

Fig. 6. Precision of the proposed smart firewall–based phishing detection model.

Fig. 7 displays the F1-Score of the proposed smart firewall model. The highest F1-Score of 98.84% was achieved using the ET classifier with the WOA algorithm, slightly outperforming ET with NoFS (98.74%) by 0.10% and with DA (98.47%) by 0.37%. The RF classifier achieved its best F1-Score of 98.81% with WOA, which was 0.27% higher than RF with NoFS (98.54%) and 0.50% higher than RF with DA (98.31%). The KNN classifier demonstrated lower overall performance, with its best F1-Score of 97.43% achieved using WOA, which was 0.17% higher than KNN with DA (97.26%) and 0.40% higher than KNN with NoFS (97.03%). A high F1-Score indicates a strong balance between precision and recall, meaning the smart firewall is effective at detecting phishing attempts while maintaining low FPRs and FNRs. This is particularly valuable in the smart firewall phishing detection, where both missing real threats and falsely blocking safe content can have significant consequences.

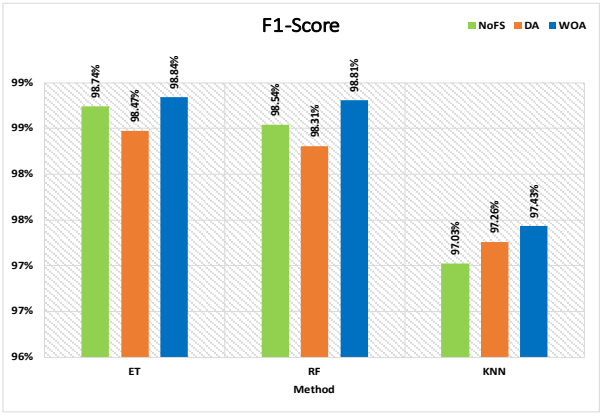

Fig. 7. F1-Score of the proposed smart firewall–based phishing detection model.

B. *Performance Comparison with Existing Approaches*

Fig. 8 displays the accuracy of the proposed firewall model, using the ET-WOA method, compared to several existing models on the ISCX-URL2016 dataset. The highest accuracy of 98.86% was achieved using the ET-WOA combination. This significantly outperformed several previous approaches, such as DNN (97.45%, a 1.41% decrease) [29], CNN (93.18%, 5.68% lower) [29], Vanilla (94.73%, 4.13% lower) [29], Sparse (95.09%, 3.77% lower) [29], and Denoising (95.71%, 3.15% lower) [29]. More advanced architectures, such as CRNN [30], also underperformed, achieving 96.85% (2.01% lower). The weakest performance was recorded by LSTM [31] at 88.25%, trailing the proposed ET-WOA by 10.61%.

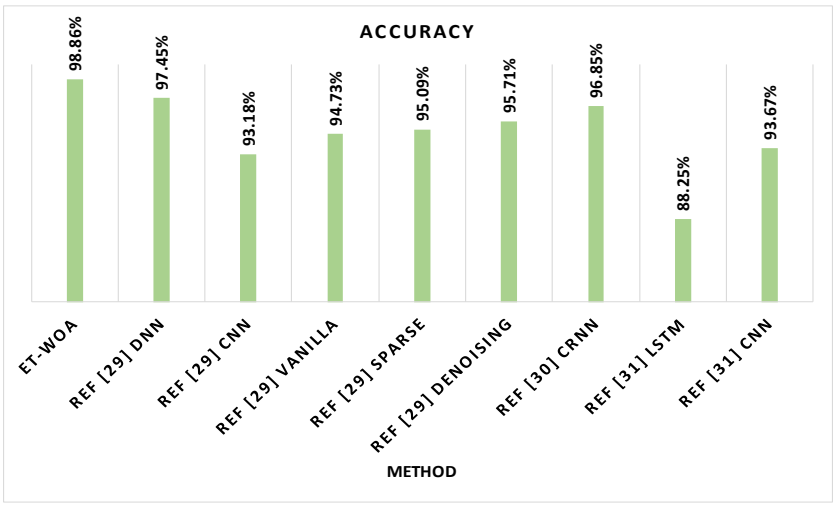

Fig. 8. Accuracy of the proposed smart firewall vs. existing methods on ISCX-URL2016.

Fig. 9 displays the accuracy of the proposed firewall model using the ET-WOA method compared against various phishing datasets and ML models. The ET-WOA model achieved the highest accuracy of 98.86%. This notably outperforms traditional classifiers, such as SVM (95.66%) [32], RF (93.22%) [32], and NB (96.01%) [36], by margins of 3.20%, 5.64%, and 2.85%, respectively. Ensemble methods and stacking classifiers reported accuracies of 92.33% (Ensemble [33]), 97.40% (Stacking1 [34]), and 97.20% (Stacking2 [34]), all of which fell short by at least 1.46% compared to the ET-WOA model. Another RF variant [35] and [36] showed 95.83% accuracy and 95.14%, which is 3.03% and 3.72 lower than ET-WOA, respectively.

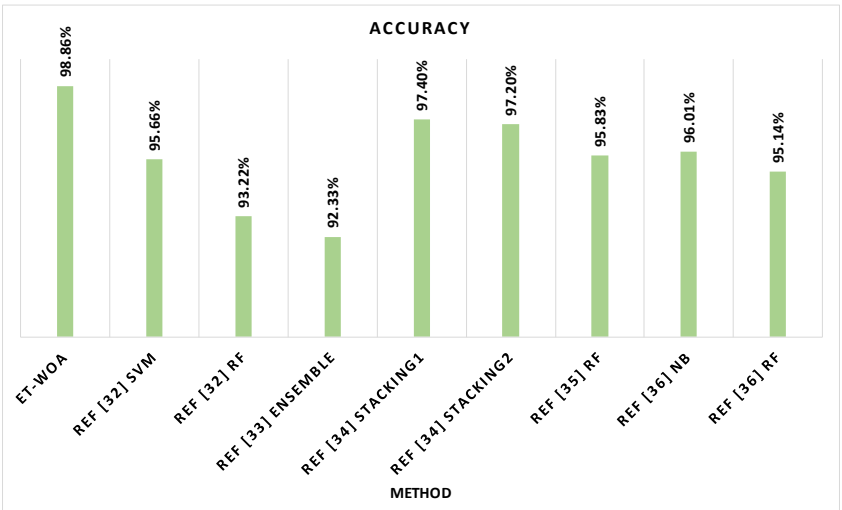

Fig. 9. Accuracy of the proposed smart firewall compared to various ML models across phishing datasets.

Fig. 10 displays the accuracy of the proposed firewall model using the ET-WOA method compared against various phishing datasets and DL models. The proposed ET-WOA achieved the highest accuracy of 98.86%, outperforming CRNN models on PhishStorm and PhishTank datasets from [30], which scored 95.05% and 94.83%, respectively. Other DL methods such as ANN (95.07% [35]), CNN (95.13%) [36], LSTM (95.14%) [36], and DBN (89.60%) [37] also reported lower accuracies, with DBN performing the worst, trailing by 9.26%. The proposed ET-WOA model's superior accuracy highlights its effectiveness and reliability for phishing detection tasks.

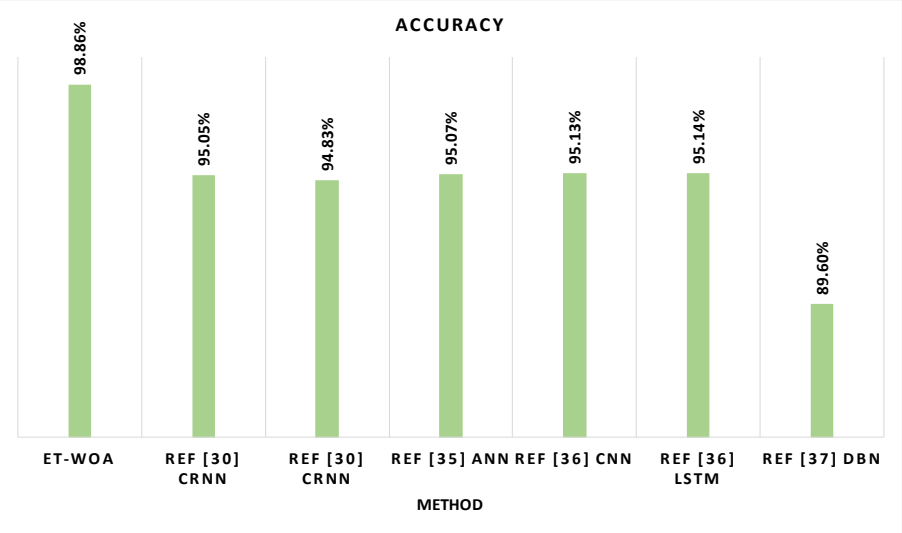

Fig. 10. Accuracy of the proposed smart firewall compared to various DL models across phishing datasets.

VI. Conclusion

This study presents a novel smart firewall model for phishing detection that harnesses bio-inspired optimization algorithms to enhance the accuracy and efficiency of threat identification. By using the WOA and DA algorithms for feature selection with robust ML classifiers, the proposed model effectively reduces data dimensionality while preserving critical phishing indicators. Experimental validation on the ISCX-URL2016 dataset demonstrates that the smart firewall powered by the WOA-ET combination achieves superior detection performance, with an accuracy of 98.86%, outperforming alternative approaches and contemporary benchmarks. These findings underscore the potential of combining bio-inspired algorithms with ML to develop intelligent, adaptive firewalls capable of proactively mitigating phishing threats.

The findings of this study are derived from experiments conducted on a single publicly available dataset (ISCX-URL2016), which may not reflect the full diversity of phishing attack techniques encountered in real-world scenarios. The evaluation considered only two metaheuristic feature selection algorithms—WOA and DA—and focused on three machine learning classifiers. Moreover, the experiments were performed in an offline environment without processing live network traffic, which may behave differently in operational deployments. These aspects should be taken into account when interpreting the results.

Future research will focus on extending the evaluation to newer and more diverse phishing datasets to test the adaptability of the proposed approach to evolving attack patterns. Deployment in real-world environments will be explored, including integration with network-level defense mechanisms such as intrusion prevention systems and smart firewalls. Additionally, ensemble feature selection strategies — such as union (∪) and intersection (∩) of features selected by multiple metaheuristic algorithms—will be investigated to assess whether combining algorithm outputs can further enhance phishing detection performance.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Resources, Supervision, MMA; Methodology, MMA, SNA, AH, and MSD; Software, AAA and MSD; Validation, AH and MSD; Formal Analysis, Writing—Original Draft Preparation, AAA, AH, and SNA; Investigation, SNA and AAA; Data Curation, AH and SNA; Writing—Review & Editing, MMA, MSD, and SNA; Visualization, AAA; all authors had approved the final version.

References

[1] K. Bennouk, N. A. Aali, Y. E. B. E. Idrissi *et al.,* "A comprehensive review and assessment of cybersecurity vulnerability detection methodologies," *J. Cybersecur. Priv.*, vol. 4, no. 4, pp. 853–908, 2024. doi: 10.3390/jcp4040040

[2] D. Dave, G. Sawhney, P. Aggarwal *et al.,* "The new frontier of cybersecurity: Emerging threats and innovations," in *Proc. 2023 29th International Conf. on Telecommunications (ICT)*, 2023, pp. 1–6.

[3] A. Petrosyan. (Jun 2025). Annual cost of cybercrime worldwide 2018–2029. *Statista*. [Online]. Available: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide

[4] Z. Alkhalil, C. Hewage, L. Nawaf *et al.,* "Phishing attacks: A recent comprehensive study and a new anatomy," *Front. Comput. Sci.*, vol. 3, 563060, 2021. doi: 10.3389/fcomp.2021.563060

[5] M. Khalil. (Apr. 2025). Phishing statistics: How AI, behavior, and first-party data are redefining cyber defense. *DeepStrike Blog*, [Online]. Available: https://deepstrike.io/blog/Phishing-Statistics-2025

[6] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms," *Knowl. Inf. Syst.*, vol. 62, pp. 1–30, 2025. doi: 10.1007/s10115-025-02429-y

[7] T. Ahmad, "AI-driven dynamic firewall optimization using reinforcement learning for anomaly detection and prevention," arXiv preprint, arXiv:2506.05356, 2025.

[8] M. M. Abualhaj, Q. Y. Shambour, A. A. Abu-Shareha *et al.,* "Enhancing malware detection through self-union feature selection using gray wolf optimizer," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 37, no. 1, pp. 197–205, 2025. doi: 10.11591/ijeecs.v37.i1.pp197-205

[9] J. Heino, A. Hakkala, and S. Virtanen, "Study of methods for endpoint-aware inspection in a next-generation firewall," *Cybersecur.*, vol. 5, 25, 2022. doi: 10.1186/s42400-022-00127-8

[10] K. Scarfone and P. Hoffman. (Sep 2009). Guidelines on firewalls and firewall policy. *National Institute of Standards and Technology*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf

[11] M. Sakhai and M. Wielgosz, "Modern cybersecurity solution using supervised machine learning," arXiv preprint, arXiv:2109.07593, 2021.

[12] M. M. Abualhaj, M. O. Hiari, A. Alsaaidah *et al.,* "Comparative analysis of whale and Harris Hawks optimization for feature selection in intrusion detection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 37, no. 1, pp. 179–185, 2025. doi: 10.11591/ijeecs.v37.i1.pp179-185

[13] M. A. Alsaleh, L. F. Mohammed, and A. A. Al-Qutayri, "Leveraging stacking machine learning models and optimization for network intrusion detection: Addressing big data challenges," *Sci. Rep.*, vol. 15, no. 1, 2025. doi: 10.1038/s41598-025-01052-9

[14] M. M. Abualhaj, A. A. Abu-Shareha, S. N. Alkhatib *et al.,* "Detecting spam using Harris Hawks optimizer as a feature selection algorithm," *Bull. Electr. Eng. Inform.*, vol. 14, no. 3, pp. 2361–2369, 2025. doi: 10.11591/eei.v14i3.9198

[15] Z.-H. Cheng, H. Shang, and C. Qian, "Detection-rate-emphasized multi-objective evolutionary feature selection for network intrusion detection," arXiv preprint, arXiv:2406.09180, 2024.

[16] S. L. Kopecky and C. Dwyer, "Nature-inspired metaheuristic techniques of firefly and grey wolf algorithms implemented in phishing intrusion detection systems," in *Proc. the 2023 Computing Conf.*, 2023, pp. 1309–1332. doi: 10.1007/978-3-031-37717-4_87

[17] G. Alshammari, M. Alshammari, T. S. Almurayziq *et al.,* "Hybrid phishing detection based on automated feature selection using the chaotic dragonfly algorithm," *Electronics*, vol. 12, no. 13, 2823, 2023. doi: 10.3390/electronics12132823

[18] O. Almomani, A. Alsaaidah, A. A. A. Shareha *et al.,* "Enhance URL defacement attack detection using particle swarm optimization and machine learning," *J. Comput. Cogn. Eng.*, 2025. doi: 10.47852/bonviewJCCE52024668

[19] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari *et al.,* "Detecting malicious URLs using lexical analysis," in *Proc. International Conf. on Network and System Security.*, 2016, pp. 467–482.

[20] Q. A. A. Haija and M. A. Fayoumi, "An intelligent identification and classification system for malicious uniform resource locators (URLs)," *Neural Comput. Appl.*, vol. 35, no. 23, pp. 16995–17011, 2023. doi: 10.1007/s00521-023-08592-z

[21] M. H. Nadimi-Shahraki, H. Zamani, Z. A. Varzaneh *et al.,* "A systematic review of the whale optimization algorithm: Theoretical foundation, improvements, and hybridizations," *Arch. Comput. Methods Eng.*, vol. 30, pp. 4113–4159, 2023. doi: 10.1007/s11831-023-09928-7

[22] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, pp. 1053–1073, 2016. doi: 10.1007/s00521-015-1920-1

[23] M. M. Abualhaj, A. A. Abu Shareha, Q. Y. Shambour *et al.,* "Customized K nearest neighbors' algorithm for malware detection," *Int. J. Data Netw. Sci.*, vol. 8, no. 1, pp. 431–438, 2024. doi: 10.5267/j.ijdns.2023.9.012

[24] Y. Alraba'nah and W. Toghuj, "A deep learning-based architecture for malaria parasite detection," *Bull. Electr. Eng. Inform.*, vol. 13, no. 1, pp. 292–299, 2024. doi: 10.11591/eei.v13i1.5485

[25] L. A. Dabbas and A. A. A. Shareha, "Early detection of female type 2 diabetes using machine learning and oversampling techniques," *J. Appl. Data Sci.*, vol. 5, no. 3, pp. 1237–1245, 2024. doi: 10.47738/jads.v5i3.298

[26] M. Imani, A. Beikmohammadi, and H. R. Arabnia, "Comprehensive analysis of random forest and XGBoost performance with SMOTE, ADASYN, and GNUS under varying imbalance levels," *Technologies*, vol. 13, no. 3, 88, 2025. doi: 10.3390/technologies13030088

[27] F. Afshar, S. S. S. Seyedabrishami, and S. Moridpour, "Application of extremely randomised trees for exploring influential factors on variant crash severity data," *Sci. Rep.*, vol. 12, 11476, 2022. doi: 10.1038/s41598-022-15693-7

[28] M. A. Raj, M. A. Thinesh, S. S. M. Varmann *et al.,* "Ensemble-based phishing website detection using extra trees classifier," *AVE Trends Intell. Comput. Syst.*, vol. 1, no. 3, pp. 142–156, 2024.

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

[30] S.-J. Bu and H.-J. Kim, "Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection," *Electronics*, vol. 11, no. 7, 1090, 2022.

[31] E. Uçar, M. Ucar, and M. O. İncetaş, "A deep learning approach for detection of malicious URLs," in *Proc. 6th Int. Manag. Inf. Syst. Conf. (IMISC)*, 2019, pp. 12–20.

[32] T. M. Rashid, M. W. Nisar, and T. Nazir, "Phishing detection using machine learning technique," in *Proc. 1st Int. Conf. Smart Syst. Emerging Technol. (SMARTTECH)*, 2020, pp. 43–46.

[33] H. Alqahtani and A. Abu-Khadrah, "Enhance the accuracy of malicious uniform resource locator detection based on effective machine learning approach," *Bull. Electr. Eng. Inform.*, vol. 13, no. 6, pp. 4422–4429, 2024. doi: 10.11591/eei.v13i6.7797

[34] A. Zamir, H. U. Khan, T. Iqbal *et al.,* "Phishing website detection using diverse machine learning algorithms," *Electron. Libr.*, vol. 38, no. 1, pp. 65–80, 2020.

[35] M. A. Daniel, S.-C. Chong, L.-Y. Chong *et al.,* "Optimising phishing detection: A comparative analysis of machine learning methods with feature selection," *J. Inf. Web Eng.*, vol. 4, no. 1, pp. 200–212, 2025. doi: 10.33093/jiwe.2025.4.1.15

[36] 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.

[37] P. Yi, Y. Guan, F. Zou *et al.,* "Web phishing detection using a deep learning framework," *Wireless Commun. Mobile Comput.*, vol. 2018, 4678746, 2018. doi: 10.1155/2018/4678746

[38] M. M. Abualhaj, S. Al-Khatib, M. O. Hiari *et al.,* "Enhancing spam detection using hybrid of Harris Hawks and Firefly optimization algorithms," *J. Soft Comput. Data Min.*, vol. 5, no. 2, 2024. doi: 10.30880/jscdm.2024.05.02.012

[39] I. Qaddara and Y. Alraba'nah, "Enhancing requirements classification using machine learning techniques," *SN Comput. Sci.*, vol. 6, no. 6, 2025. doi: 10.1007/s42979-025-04158-z

[40] Y. Sanjalawe, S. Fraihat, S. R. Al-E'Mari *et al.,* "Hybrid deep learning for human fall detection: A synergistic approach using YOLOv8 and time–space transformers," *IEEE Access*, 2025. doi: 10.1109/ACCESS.2025.3547914

[41] Y. Sanjalawe, S. Al-E'mari, S. Fraihat *et al.,* "A deep learning-driven multi-layered steganographic approach for enhanced data security," *Sci. Rep.*, vol. 15, no. 1, 4761, 2025. doi: 10.1038/s41598-025-89189-5