JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 5 No. 2 (2024) 161-174
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Enhancing Spam Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms

Mosleh M. Abualhaj[1]*, Sumaya Al-Khatib[1], Mohammad O. Hiari[1], Qusai Y. Shambour[1]

[1] *Faculty of Information Technology*
Al-Ahliyya Amman University, Amman, 19111, JORDAN

*Corresponding Author: m.abualhaj@ammanu.edu.jo

DOI: https://doi.org/10.30880/jscdm.2024.05.02.012

Article Info

Received: 27 July 2024
Accepted: 18 October 2024
Available online: 18 December 2024

Keywords

Spam, machine learning, feature selection, firefly algorithm, Harris Hawks algorithm

Abstract

Email spam is a significant challenge that negatively affects communication and data security. Machine Learning (ML) is a widely adopted approach for spam detection. However, the large amount of spam data can degrade ML models performance. To address this issue, this paper proposes a novel feature selection method that combines the Firefly Optimization Algorithm (FOA) and Harris Hawks Optimization (HHO) to enhance the effectiveness of ML models in spam detection. The proposed method was evaluated using the ISCX-URL2016 dataset with several classifiers. Hyperparameters for these classifiers were optimized using the Grid Search technique to ensure the best possible performance. The achieved results indicate that the Extra Trees (ET) classifier, when combined with the novel FOA and HHO-based feature selection method, achieved the highest accuracy of 99.83%, outperforming all other tested classifiers. This demonstrates the potential of our approach in significantly improving spam detection systems by effectively handling large-scale spam data.

1. Introduction

Currently, email has become firmly established in our society due to extensive research efforts that have made email technology more user-friendly and cost-effective. Therefore, an email system has become a crucial method of communication for tens of millions of people since it allows for the quick and instantaneous electronic distribution of messages to anyone without incurring any costs [1], [2]. Over the past decade, email systems have been extensively overused and exploited through the proliferation of Spam. Spam refers to frequently sending unwanted messages to one or more receivers that the recipient does not desire [3]. Based on the data provided in [4], approximately 46.8% of email messages are classified as Spam. Spam is not just vexing and bothersome; it is also a chronic issue that can inflict substantial damage and adversely affect internet users and administrators.
Furthermore, the danger associated with Spam has escalated significantly, with 91% of spam messages now including a URL. This means phishing websites and Trojan infections can be accessed with only a single click [5]. Considering that Spam is both an annoyance and a danger, it is important to explore the different methods used to filter and prevent it. Email spam filtering can be accomplished using binary classification, which can be resolved using machine learning (ML) algorithms and various anti-spam tools that employ different algorithmic approaches [6].

ML is a field of research that focuses on utilizing computers to replicate human learning processes. It involves computers acquiring new knowledge and skills, recognizing current knowledge, constantly enhancing performance, and using self-improvement techniques. In ML, the two primary factors to consider are generally a trade-off between speed and accuracy. Conventional ML models necessitate a human operator to create features that accurately depict the significant characteristics of the objects. These models are challenging or laborious to apply to different types of data [7], [8]. An effective ML model necessitates increasing the amount of high-dimensional data for training, resulting in a longer training period.

Dimensionality reduction is a commonly used technique in preprocessing high-dimensional data for analysis, visualization, and modeling. A straightforward method for reducing dimensionality is feature selection, where only the input dimensions that hold pertinent information for solving the problem are chosen. Feature selection approaches can be employed individually or in combination to enhance performance, including estimated accuracy, visualization, and comprehensibility of acquired knowledge [9] – [11]. In general, features can be classified as relevant, irrelevant, or redundant. During the feature selection phase, the ML algorithm chooses a subset of feature data. The optimal subset has the fewest dimensions that have the greatest impact on improving learning accuracy. Metaheuristic optimization algorithms are highly effective tools for selecting features for ML. These algorithms are specifically developed to identify the most effective, or nearly most effective, solutions to intricate issues [9] – [11].

In this paper, a novel ML model, which employs two of the robust metaheuristic methods for feature selection, will be designed for Spam email detection. These two methods are the Firefly Optimization Algorithm (FOA) and Harris Hawks Optimization (HHO) [9], [12]. Therefore, the designed model is called Spam-FOA-HHO. In addition, the designed model will be evaluated using various well-known ML classifiers with custom parameter settings. These classifiers are Extremely Randomized Trees (Extra Trees [ET]), Decision Tree (DT), AdaBoost, Support Vector Machine (SVM), and Logistic Regression (LR) [13] – [15].

2. Background

The section discusses the ISCX-URL2016 dataset and feature selection algorithms that have been used in the designed Spam-FOA-HHO model.

2.1 ISCX-URL2016 Dataset

The ISCX-URL-2016 dataset from the Canadian Institute for Cyber Security contains four different types of malicious URLs, namely Spam, Phishing, Malware, and Defacement, besides benign URLs. The ISCX-URL-2016 dataset is divided into four datasets, one for each URL type. The ISCX-URL-2016 Spam dataset, which is the concern of this research, contains 79 features and 14479 samples, balanced between Spam and benign samples. The data within the features are distributed in wide ranges, and many of the values are Null [16,17]. Therefore, the ISCX-URL-2016 Spam dataset requires pre-processing to be prepared for the ML model. Table 1 shows the features of the ISCX-URL2016 Spam dataset.

Table1 *ISCX-URL2016 Spam dataset*

#	Feature	Min	Max	#	Feature	Min	Max
1	Querylength	0	1385	41	Directory_DigitCount	-1	46
2	domain_token_count	2	5	42	File_name_DigitCount	-1	42
3	path_token_count	0	68	43	Extension_DigitCount	-1	236
4	avgdomaintokenlen	2	13	44	Query_DigitCount	-1	236
5	longdomaintokenlen	2	23	45	URL_Letter_Count	15	1202
6	avgpathtokenlen	1.5	65	46	host_letter_count	2	44
7	tld	2	5	47	Directory_LetterCount	-1	125
8	charcompvowels	0	193	48	Filename_LetterCount	-1	186
9	charcompace	0	142	49	Extension_LetterCount	-1	1179
10	ldl_url	0	207	50	Query_LetterCount	-1	1173
11	ldl_domain	0	1	51	LongestPathTokenLength	0	1393
12	ldl_path	0	207	52	Domain_LongestWordLength	2	23
13	ldl_filename	0	15	53	Path_LongestWordLength	0	44
14	ldl_getArg	0	207	54	sub-Directory_LongestWordLength	-1	24
15	dld_url	0	31	55	Arguments_LongestWordLength	-1	79
16	dld_domain	0	0	56	URL_sensitiveWord	0	3
17	dld_path	0	31	57	URLQueries_variable	0	13
18	dld_filename	0	10	58	spcharUrl	1	16

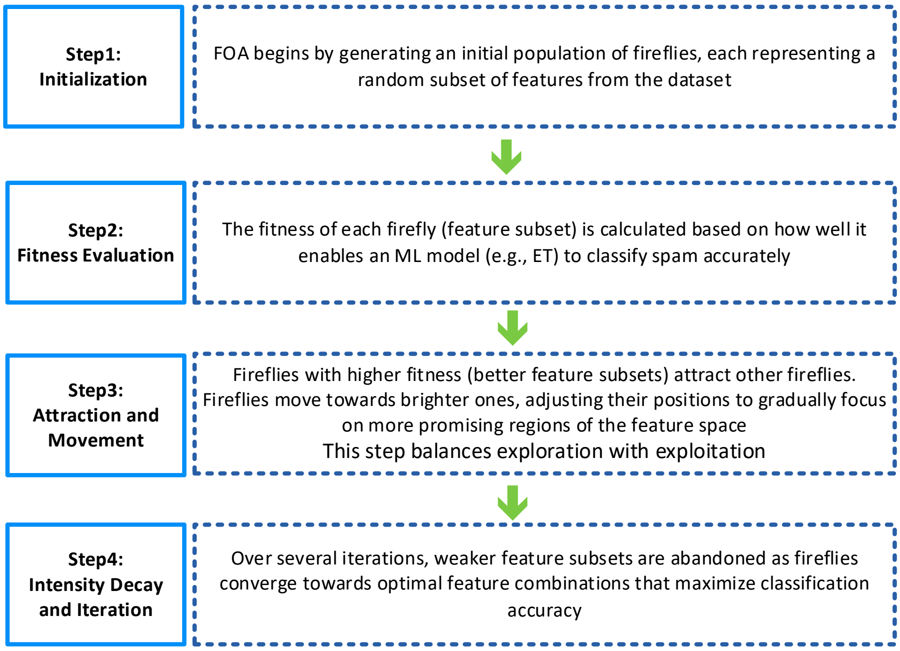
19	dld_getArg	0	31	59	delimeter_Domain	0	4
20	urlLen	22	1424	60	delimeter_path	0	64
21	domainlength	5	49	61	delimeter_Count	-1	29
22	pathLength	1	1402	62	NumberRate_URL	0	0.513
23	subDirLen	1	1402	63	NumberRate_Domain	0	0.5
24	fileNameLen	1	165	64	NumberRate_DirectoryName	-1	0.833
25	this.fileExtLen	1	5	65	NumberRate_FileName	-1	1
26	ArgLen	1	1388	66	NumberRate_Extension	-1	1
27	pathurlRatio	0.026	0.9845	67	NumberRate_AfterPath	-1	1
28	ArgUrlRatio	0.005	0.9747	68	SymbolCount_URL	3	37
29	argDomanRatio	0.040	92.533	69	SymbolCount_Domain	1	4
30	domainUrlRatio	0.010	0.7894	70	SymbolCount_Directoryname	-1	19
31	pathDomainRatio	0.033	93.466	71	SymbolCount_FileName	-1	31
32	argPathRatio	0.006	2	72	SymbolCount_Extension	-1	30
33	executable	0	0	73	SymbolCount_Afterpath	-1	29
34	isPortEighty	-1	-1	74	Entropy_URL	0.49	0.895
35	NumberofDotsinURL	1	19	75	Entropy_Domain	0.68	1
36	ISIpAddressInDomainName	-1	-1	76	Entropy_DirectoryName	-1	0.962
37	CharacterContinuityRate	0.2	1	77	Entropy_Filename	-1	1
38	LongestVariableValue	-1	1385	78	Entropy_Extension	-1	1
39	URL_DigitCount	0	236	79	Entropy_Afterpath	-1	1
40	host_DigitCount	0	4				

2.2 Feature Selection Algorithms

This subsection discusses the feature selection algorithms that have been used in the designed Spam-FOA-HHO model, namely HHO and FOA. The HHO and FOA algorithms are advantageous due to their efficient convergence, which allows them to quickly find optimal or near-optimal solutions. They balance exploration and exploitation, ensuring thorough search space coverage and effective solution refinement.

2.2.1 FOA Algorithm

The FOA is a novel metaheuristic swarm optimization algorithm that draws inspiration from the innate behavior of fireflies. The inherent behavior of fireflies is predicated on the phenomena of bioluminescence. Fireflies emit short and regular bursts of light to communicate with other fireflies and lure possible prey. The bioluminescent emission of fireflies can be mathematically modeled based on the objective function that needs to be optimized, which allows for the formulation of optimization algorithms. The core structure of the FOA algorithm adheres to three idealized rules: 1) All fireflies are unisex; consequently, a firefly will be drawn to other fireflies regardless of their sex. 2) The level of attraction in fireflies is directly related to their luminosity. Therefore, when two fireflies are flashing, the one with lower brightness will approach the one with higher brightness. Moreover, the level of appeal is directly correlated with the brightness level, therefore diminishing as the distance rises. If no firefly is brighter than a specific one, it will move randomly. The nature of an objective function determines the brightness of each firefly. When maximizing the objective function, the firefly with the higher value is considered brighter. Conversely, when minimizing the objective function, the firefly with the lower value is considered brighter [12], [18], [19]. Fig. 1 illustrates the mechanism and steps employed by the FOA for feature selection.



Step1: Initialization

FOA begins by generating an initial population of fireflies, each representing a random subset of features from the dataset

Step2: Fitness Evaluation

The fitness of each firefly (feature subset) is calculated based on how well it enables an ML model (e.g., ET) to classify spam accurately

Step3: Attraction and Movement

Fireflies with higher fitness (better feature subsets) attract other fireflies. Fireflies move towards brighter ones, adjusting their positions to gradually focus on more promising regions of the feature space

This step balances exploration with exploitation

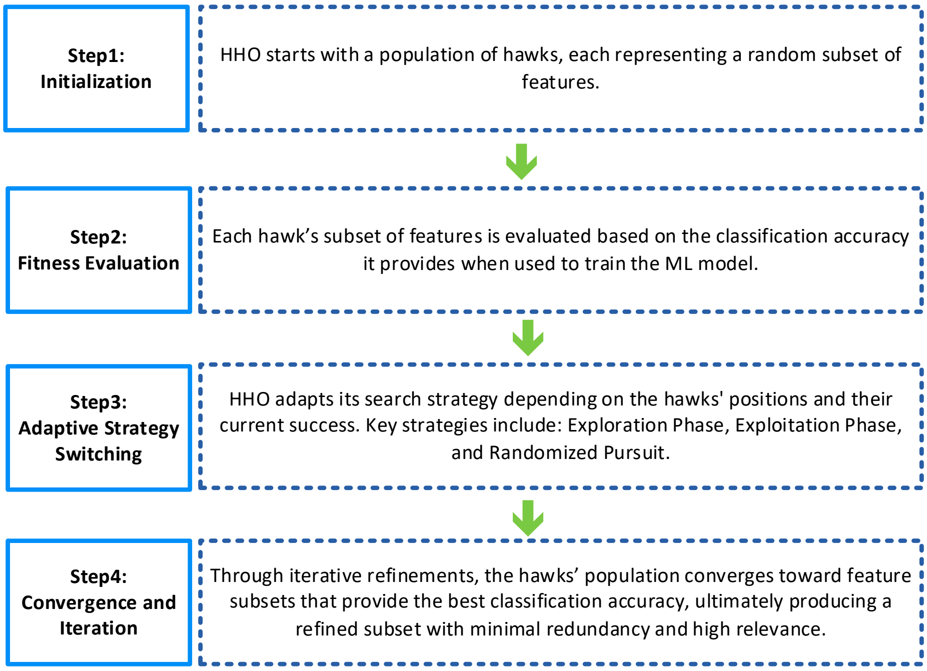
Step4: Intensity Decay and Iteration

Over several iterations, weaker feature subsets are abandoned as fireflies converge towards optimal feature combinations that maximize classification accuracy

Fig. 1 *Steps employed by the FOA for feature selection*

2.2.2 HHO Algorithm

HHO is a metaheuristic swarm optimization method that models the actions of Harris hawks when they look for and hunt rabbits. The Harris hawks are considered to be one of the most intelligent birds in nature due to their observed innovative feeding techniques. Throughout the whole hunting procedure for a rabbit, Harris hawks exhibit cooperative behavior and demonstrate intelligent decision-making based on their observations. Their hunting habits against rabbits can be analytically characterized using five different approaches and two processes for exploring and exploiting. Moreover, the HHO method outperforms other algorithms in numerical benchmark tests because it effectively balances exploitation and exploration. This results in the acquisition of high-quality solutions and an accelerated convergence rate. HHO is also a population-based algorithm. A group of possible vectors constitutes a population. A population is composed of several vectors, with each vector representing a candidate in the population. The population is randomly initialized inside the design space using a uniform distribution. Following startup, each member of the population adjusts its position according to distinct phases [9], [20], [21]. Fig. 2. illustrates the mechanism and steps employed by the HHO for feature selection.



Step1: Initialization

HHO starts with a population of hawks, each representing a random subset of features.

Step2: Fitness Evaluation

Each hawk's subset of features is evaluated based on the classification accuracy it provides when used to train the ML model.

Step3: Adaptive Strategy Switching

HHO adapts its search strategy depending on the hawks' positions and their current success. Key strategies include: Exploration Phase, Exploitation Phase, and Randomized Pursuit.

Step4: Convergence and Iteration

Through iterative refinements, the hawks' population converges toward feature subsets that provide the best classification accuracy, ultimately producing a refined subset with minimal redundancy and high relevance.

Fig. 2 *Steps employed by the HHO for feature selection*

3. Related Works

Numerous research publications focus on the identification, classification, and filtering of email spam. Zhou et al. [22] suggest a cost-sensitive approach to filtering spam emails using a three-way strategy that combines Bayesian analysis, thresholding, and probability assessment. This approach has the potential to decrease the rate of incorrectly categorizing a valid email as spam and provide superior performance in terms of cost-sensitive factors. The dataset was divided into two portions, namely the training set and the testing set, with a distribution of 80% and 20%, respectively. The proposed approach attained a maximum accuracy of 93.94%.

Rathi et al. [23] investigate various data mining techniques for spam datasets with the objective of determining the optimal classifier for email classification. The authors evaluate the effectiveness of several classifiers by comparing their performance with and without the use of feature selection algorithms. Initially, they experiment with the entire dataset without picking the specific characteristics, applying classifiers individually, and evaluating the outcomes. At that point, they employ the Best-First peculiarity selection algorithm to identify the desired features, after which they apply various classifiers for classification. Research has shown that incorporating the feature selection procedure within the experiment leads to enhanced accuracy in the outcomes. We ultimately determined that Random Tree was the most effective classifier for identifying spam mail.

Zhang et al. [24] introduce a binary search technique called MB-PSO, which combines particle swarm optimization with a mutation operator. The process commences with a wrapper-based technique for selecting features using the C4.5 decision tree classifier and weighting factors. The proposed strategy was able to achieve an accuracy rate of 94.27%. Idris et al. also used PSO to improve random detector production [25]. During the random detector generation step of the negative selection procedure, the algorithm creates detectors. The proposed method uses a local outlier factor (LOF) as a fitness function to generate the detector. After generating the detectors, a distance measure and a threshold value are used to improve the differentiation between non-spam and spam. The proposed approach demonstrated an accuracy rate of 83.20%.

Mamun et al. [26] introduced a recently published dataset named ISCX-URL-2016, which contains malicious URLs that are accessible to the public. In addition, Mamun et al. conducted experiments using RF, C4.5, and kNN algorithms to identify and categorize malicious URLs in this new dataset. They generated features directly from the URL, including URL length, domain entropy, and arguments. Regarding identifying harmful URLs, the RF model had the highest precision and recall, surpassing 99%. On the other hand, both the C4.5 and kNN models achieved precision and recall rates above 97%. Random Forest (RF) demonstrated superior performance in classifying malicious communications, with a recall and precision of 97%. During the second classification phase, all models exhibit precision and recall values ranging from 92% to 97%. Ultimately, the authors proved that the act of obscuring harmful website addresses does indeed lower the accuracy of the ML models' ability to detect and classify them.

Cui et al. [27] introduced a system that utilizes NB, DT, and SVM classifiers to identify malicious URLs. The stated accuracy for all classifiers exceeds 98.7%. Thanks to the exceptional performance of the models, the authors assert that they have implemented the system, examining a daily volume of data amounting to 2 TB. Zhao et al. [28] expanded on this study by conducting a comparison between the performance of RF and a Gated Recurrent Neural Network (GRU) in detecting malicious URLs. They employed the same dataset from [27]. Zhao et al. discovered that the GRU model exhibited superior performance compared to the RF model, achieving an accuracy of 98.5% and 96.4%, respectively. Training datasets ranging from 600 to 240,000 samples showed this pattern. In addition, they displayed graphs illustrating the distributions of each categorization (Legitimate, SQL Injection, XSS Attack, Sensitive File Attack, and Directory Traversal) based on the "Number of Characters" feature. Further statistical analysis of each feature can provide greater insight into the challenge of detecting malicious URLs, which is a significant contribution to this research study.

Vinayakumar et al. [29] tested how well different deep learning architectures (LSTM, CNN, bidirectional recurrent structures, LSTM-CNN hybrids, and stacked CNNs) worked at finding malicious URLs and compared them to the architecture they came up with. The architectures investigated were executed using Keras, as detailed in the Frameworks section. The dataset consists of two parts and incorporates data from Alexa.com, DMOZ Directory, Sophos, and other relevant sources. The models achieve an accuracy range of 95% to 99% on the initial dataset, with LSTM being the most effective model. The authors point out the difference between the random-split and time-split versions of the dataset, showing that the accuracy results of the time-split version are more variable than those of the random-split version of the second dataset. The models' accuracy in the random-split component of the second dataset varies from 95% to 96.6%, whereas in the time-split version, the models have accuracy ranging from 93% to 97.1%.

Hung et al. [30] conducted a series of deep-learning experiments using CNN to train the model in URL embedding. The VirusTotal anti-virus organization uses a substantial dataset. The team endeavored to uncover the limits of detecting dangerous URLs based on lexical features, such as the inability to discern the correct semantic meaning and patterns in URLs. To tackle this issue, they employed an end-to-end deep learning architecture. The researchers tackled the issue of determining the maliciousness of a URL by framing it as a binary classification task. The researchers use a character-level CNN technique to gain knowledge about the frequency and order of characters in the URL. Additionally, they utilize a word-level CNN to identify distinct terms within the URL. The primary objective of the paper was to employ character-level and word-level CNNs to overcome the shortcomings of earlier approaches and accurately detect dangerous URLs.

4. Proposed Spam-FOA-HHO Model

This section outlines the proposed Spam-FOA-HHO model for detecting Spam emails. First, the ISCX-URL2016 dataset will be processed to be prepared for training and testing the proposed Spam-FOA-HHO model. Then, the proposed feature selection method that will be used in the proposed model will be discussed. After that, the specific settings of the classifiers utilized by the Spam-FOA-HHO model will be detailed. Finally, the evaluation mechanism of the proposed model will be briefed. Fig. 3. illustrates the proposed Spam-FOA-HHO model.

4.1 ISCX-URL2016 Dataset Preprocessing

As mentioned in Section 2.1, the ISCX-URL2016 Spam dataset will be used in the evaluation process of the proposed Spam-FOA-HHO model. Several features of the ISCX-URL2016 Spam dataset contain Null values. These features are avgpathtokenlen (#6), NumberRate_Extension (#66), NumberRate_AfterPath (#67), Entropy_DirectoryName (#76), Entropy_Filename (#77), Entropy_Extension (#78), and Entropy_Afterpath (#79) [16]. During the preprocessing stage, these seven features have been removed from the ISCX-URL2016 Spam dataset. Therefore, the dataset contains 72 features rather than 79 features. In addition to Null, the features contain values distributed on a large scale, as shown in Table 1. The Spam-FOA-HHO model will use the Min-max normalization mechanism to a small scale between 0 and 1. Table 2 shows a sample of the ISCX-URL2016 Spam dataset before and after normalization. The final step in the preprocessing stage is feature selection, which is discussed in the following subsection.

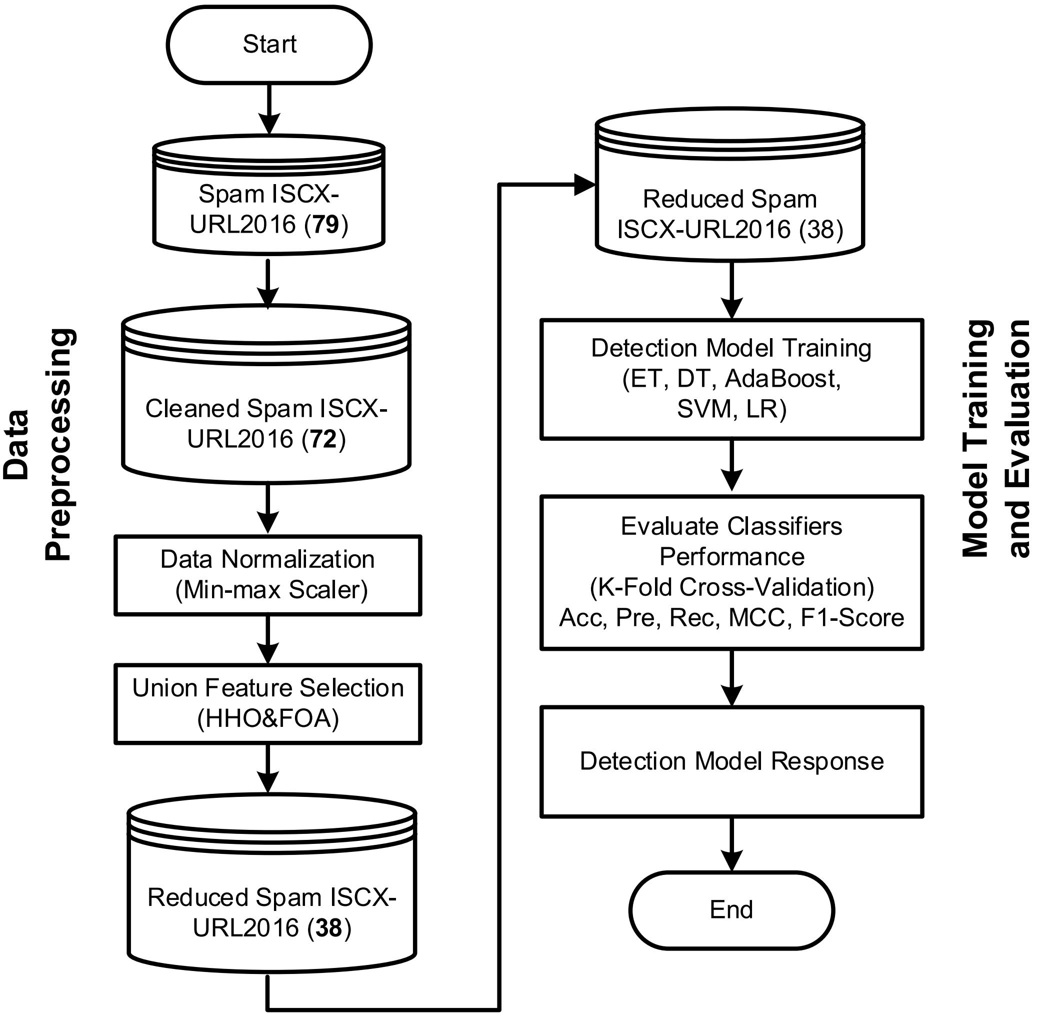

Start
Spam ISCX-URL2016 (79)
Cleaned Spam ISCX-URL2016 (72)
Data Normalization (Min-max Scaler)
Union Feature Selection (HHO&FOA)
Reduced Spam ISCX-URL2016 (38)
Data Preprocessing
Reduced Spam ISCX-URL2016 (38)
Detection Model Training (ET, DT, AdaBoost, SVM, LR)
Evaluate Classifiers Performance (K-Fold Cross-Validation) Acc, Pre, Rec, MCC, F1-Score
Detection Model Response
End
Model Training and Evaluation

Fig. 3 *Spam-FOA-HHO model*

Table 2 *Sample of the ISCX-URL2016 Spam dataset before and after normalization*

Before Normalization	After Normalization
0, 2, 5.5, 2, 7	0, 0, 0.318182, 0, 0.049296
0, 3, 5, 3, 8	0, 0.333333, 0.272727, 0.333333, 0.056338
2, 2, 4, 2, 11	0.001444, 0, 0.181818, 0, 0.077465
0, 2, 4.5, 2, 10	0, 0, 0.227273, 0, 0.070423
19, 2, 6, 2, 5	0.013718, 0, 0.363636, 0, 0.035211

4.2 Feature Selection

In order to gain a more in-depth understanding of how to develop efficient methods for identifying and classifying spam email forms, we analyze the features of the ISCX-URL-2016 Spam dataset. Once we remove the features containing null values from the dataset, we find a total of 72 features. These properties are intrinsically distinct from one another in terms of their capacity to forecast outcomes and, thus, their usefulness for an ML model. Consequently, the utilization of an appropriate algorithm for feature selection offers a number of advantages. Firstly, we efficiently reduced the ISCX-URL-2016 spam dataset to reduce the number of irrelevant features. This helps minimize the amount of computing resources required to create accurate predictions for real-world samples. Furthermore, the process of spam identification uses less time and computational resources to create irrelevant or redundant features [9,18,20]. Additionally, we will choose the features with the highest association with categorizing spam samples. This study will utilize both the FOA and the HHO for feature selection.

Typically, the feature selection process employs a single approach, like the HHO algorithm. We use this method to identify features with the most accurate prediction capabilities, thereby improving the performance of an ML model [9,18,20]. However, this study presents a novel feature selection technique that combines two well-known and robust algorithms: the FOA algorithm and the HHO algorithm. The first steps involve applying the HHO algorithm to the ISCX-URL-2016 spam dataset to choose the most effective features based on the dataset's operational behavior. In the subsequent step, we apply the FOA algorithm to the same dataset to identify the most significant aspects based on its operational mechanisms. Once the process is complete, we combine the two subsets of features selected by the HHO and FOA algorithms to create a single subset. This union subset includes the features deemed optimal by both algorithms, enabling the implementation of both the strengths of FOA and HHO. Table 3 presents the combination of properties derived from the FOA and HHO algorithms. Fig. 4. presents a representation of the union feature selection approach. This method allows for the optimization of the resulting subset of features based on the complementary strengths of the FOA and HHO algorithms, which ultimately improves the ML model's prediction performance.

Table 3 *Selected features by different methods*

Method	Selected features (feature #)
FOA	0,1,3,5,7,9,11,12,19,21,22,25,28,30,32,35,38,39,41,42,44,46,47,49,51,53,59,67,68,70,71
HHO	5,17,25,26,29,33,35,46,61,62
Union of FOA & HHO	0,1,3,5,7,9,11,12,17, 19,21,22,25,26,28,29,30,32,33,35,38,39,41,42,44,46,47,49,51,53,59,61,62,67,68,70,71

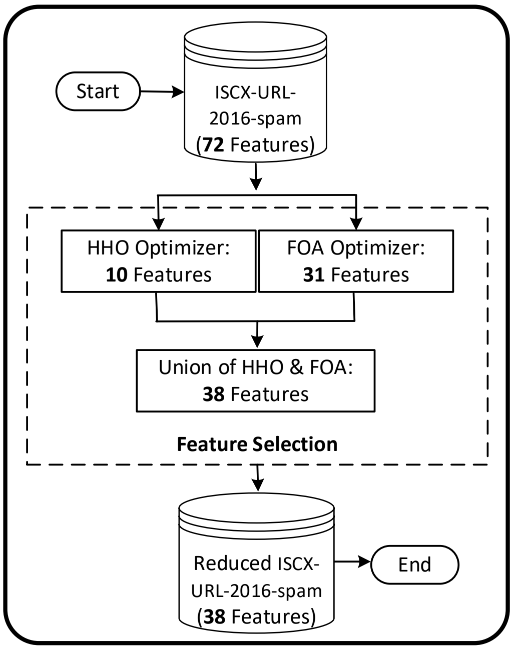

Start
ISCX-URL-2016-spam (72 Features)
HHO Optimizer: 10 Features
FOA Optimizer: 31 Features
Union of HHO & FOA: 38 Features
Feature Selection
Reduced ISCX-URL-2016-spam (38 Features)
End

Fig. 4 *The proposed union feature selection method*

4.3 Classification

The designed Spam-FOA-HHO model will be evaluated using Extra Trees (ET), Decision Tree (DT), AdaBoost, Support Vector Machine (SVM), and Logistic Regression (LR) classifiers. The hyperparameters of these classifiers are optimized to enhance the performance of the Spam-FOA-HHO model for spam detection. The assigned hyperparameter values are determined using the Grid Search algorithm, which is a straightforward method for hyperparameter tuning. Grid Search works by dividing the domain of the hyperparameters into a discrete grid and then evaluating every combination of values within this grid. Performance metrics are calculated using cross-validation, and the combination of hyperparameters that maximizes the average value in cross-validation is identified as the optimal set [31], [32]. These resulting hyperparameter values are tailored for the ISCX-URL-2016 spam dataset, which contains 14,479 samples and 72 features.

Optimizing the hyperparameters of classifiers for spam detection requires careful tuning to balance the model's complexity, accuracy, and generalization ability. The technique that demonstrates the best performance will be identified for the proposed Spam-FOA-HHO model. Table 4 displays the key parameters for each of the classifiers.

Table 4 *Classifiers hyperparameters*

Classifier	Hyperparameters	Description	Assigned value
DT and ET	criterion	Function to measure the quality of a split	gini

	max_depth	Maximum depth of the tree	DT = “30”
			ET =“None”

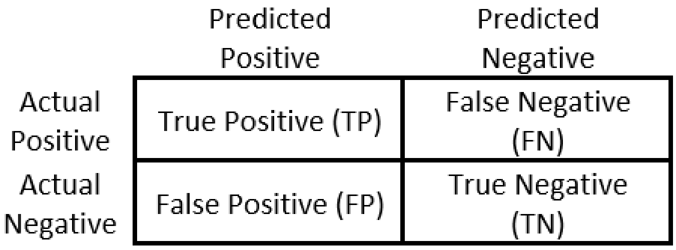
	min_samples_split	Minimum number of samples required to split an internal node.	2
	max_features	The number of features to consider when looking for the best split	sqrt
ET	n_estimators	Number of trees in the forest.	500
AdaBoost	n_estimators	The maximum number of estimators (weak learners) to be added	200
	learning_rate	Weight applied to each classifier at each boosting iteration	0.1
	base_estimator	The base estimator from which the boosted ensemble is built	DT with max_depth=1

SVM	c	Regularization parameter. It controls the trade-off between achieving a low training error and a low testing error	1.0
	kernel	Specifies the kernel type to be used in the algorithm	rbf
	gamma	Affects the distance a single training example reaches. Low values mean far, high values mean close. Higher values can lead to overfitting.	scale
LR	C	controls the amount of regularization applied to the model	1.0
	penalty	determines the type of regularization to use	l2
	solver	specifies the algorithm to use in the optimization problem	lbfgs
	max_iter	specifies the maximum number of iterations taken for the solver to converge	100

4.4 Evaluation

The designed Spam-FOA-HHO model will be evaluated using the popular K-fold cross-validation method. K-fold cross-validation divides the ISCX-URL-2016 spam dataset into k folds (subsets), which are five subsets in this work. The model is then trained and validated five times, each time using a different subset as the validation set and the remaining subsets as the training set. After completing the five tests, the average performance metrics from the five subsets are calculated, providing a comprehensive evaluation of the model's performance [33], [34].

The designed Spam-FOA-HHO model is evaluated using accuracy (Acc), precision (Pre), recall (Rec), and Matthews Correlation Coefficients (MCC). Considering all four metrics more accurate estimate of the model's generalization ability. In Fig. 5., the confusion matrix is provided to calculate the evaluation measures [33], [34]. In designed Spam-FOA-HHO model, the spam class is designated as the positive class, whereas the benign class is designated as the negative class. Each evaluation metric is described based on this assumption, and they are as follows. Acc is the ratio of accurately classified spam and ham divided by the total number classified ham, Equation (1). Pre is the ratio of the accurately classified ham divided by the total number of predicted hams, Equation (2). Rec is the number of accurately predicted spam divided by the total number of spams, Equation (3). MCC is a measure of the quality of classification with two classes, Equation (4) [33]-[35].



	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Fig. 5 *Confusion matrix*

$$Acc = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

$$Rec = \frac{TP}{(TP + FN)} \quad (2)$$

$$Pre = \frac{TP}{(TP + FP)} \quad (3)$$

$$Pre = \frac{TP}{(TP + FP)} \quad (4)$$

5. Results and Discussion

The experiments to evaluate the designed Spam-FOA-HHO model were run on a desktop with the following specifications: HP Z2 G9 Tower, Processor Intel Core i9-12900K (5.20 GHz, 30M Cache, 16 Cores), 128GB DDR5, 2TB SSD, RTX A5000 24Gb, Win 11Pro. Several Python tools from scikit-learn library were used to build and evaluate the Spam-FOA-HHO model, including: ExtraTreesClassifier, DecisionTreeClassifier, AdaBoostClassifier, SVC, LogisticRegression, train_test_split, accuracy_score, GridSearchCV.

The proposed Spam-FOA-HHO model demonstrates significant variances in accuracy across ET, DT, AdaBoost, SVM, and LR classifiers (Fig. 6.). The highest accuracies are achieved with the combined HHO and FOA features, particularly ET at 99.83%, AdaBoost at 99.76%, and SVM at 99.41%. This indicates that the union of HHO and FOA algorithms captures a more comprehensive and informative feature set. ET consistently excels due to its ensemble nature, which allows it to manage complex feature interactions and reduce overfitting, making it highly robust across all feature selection methods. DT performs slightly better with FOA (99.79%) than HHO (99.69%), suggesting a better alignment with FOA's selected features. LR shows the lowest accuracy, however, it benefits from the combined feature set (98.69%), suggesting better linear relationships.

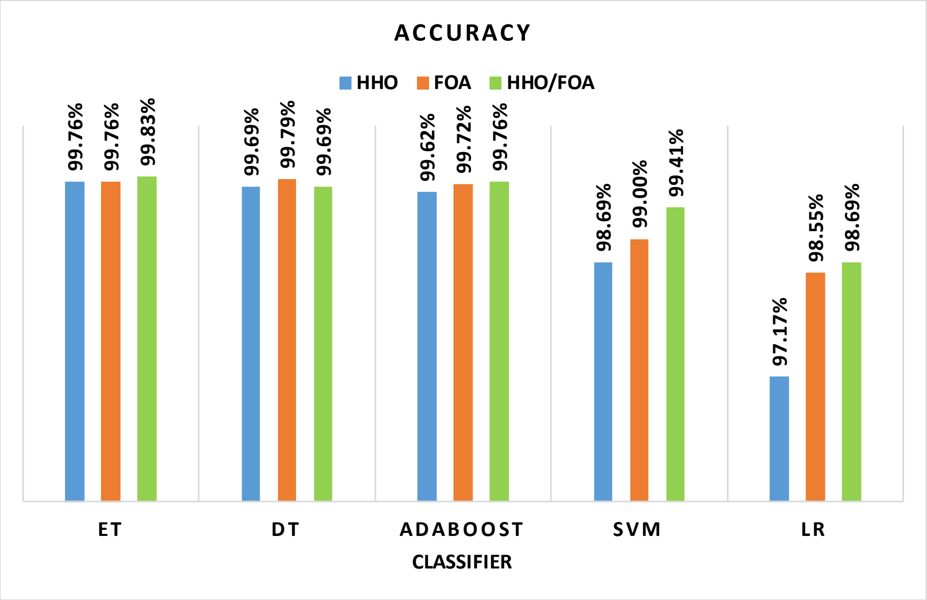


Fig. 6 *Accuracy of the proposed Spam-FOA-HHO*

Fig. 7 shows that the Spam-FOA-HHO model achieved impressive recall across different classifiers. Specifically, HHO combined with ET achieves a recall of 99.76%, DT 99.69%, AdaBoost 99.62%, SVM 98.69%, and LR 97.17%. When FOA is employed, ET reaches 99.76%, DT 99.79%, AdaBoost 99.72%, SVM 99.00%, and LR 98.55%. The union of HHO and FOA yields even higher recall rates, with ET at 99.83%, DT 99.69%, AdaBoost 99.76%, SVM 99.41%, and LR 98.69%.

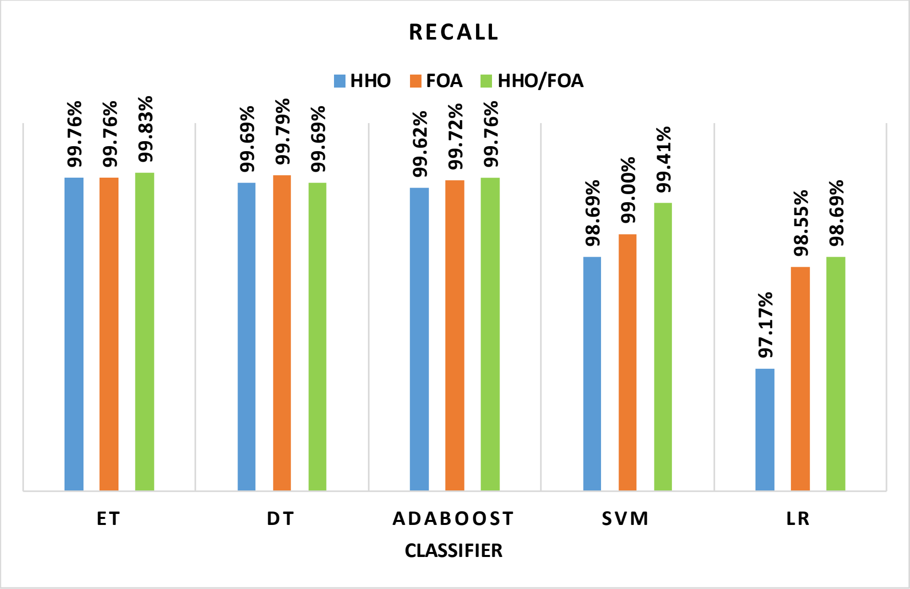


Fig. 7 *Recall of the proposed Spam-FOA-HHO*

Fig. 8 shows that the Spam-FOA-HHO model achieved impressive precision across a variety of classifiers. Specifically, with HHO, the precision achieved with ET is 99.76%, DT 99.69%, AdaBoost 99.62%, SVM 98.70%, and LR 97.31%. When utilizing FOA, the precision for ET remains at 99.76%, improves slightly with DT to 99.79%, increases with AdaBoost to 99.72%, rises with SVM to 99.01%, and enhances with LR to 98.57%. The combined use of HHO and FOA results in the highest precision values, with ET reaching 99.83%, DT maintaining 99.69%, AdaBoost achieving 99.76%, SVM improving to 99.42%, and LR increasing to 98.71%.

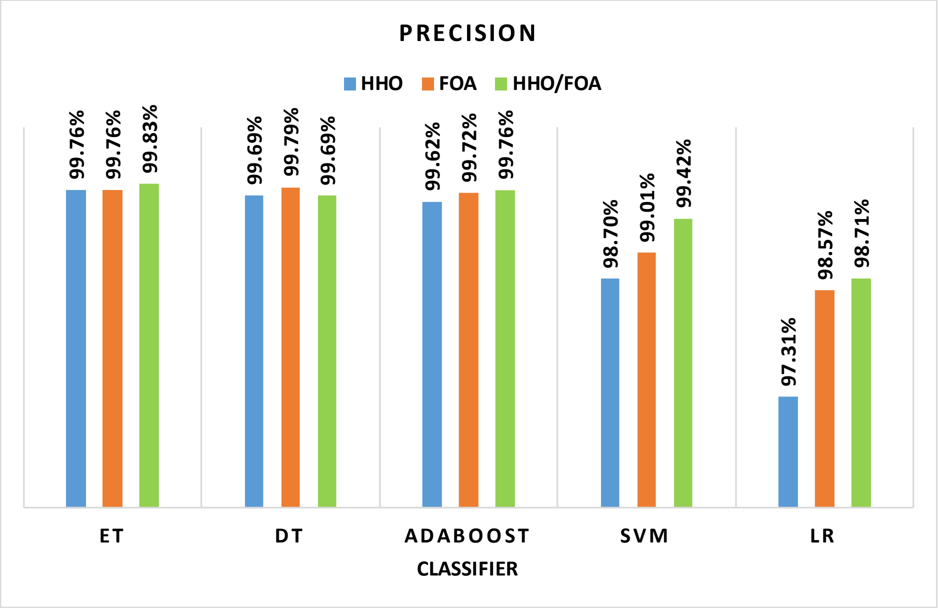


Fig. 8 *Precision of the proposed Spam-FOA-HHO*

Fig. 9 shows that the proposed Spam-FOA-HHO model achieved vary MCC across classifiers. The highest MCC is achieved with the combined HHO/FOA features, particularly by ET at 99.65%, indicating that the union of these algorithms captures a comprehensive and informative feature set, allowing ET to effectively manage complex interactions and reduce overfitting, leading to highly accurate and balanced predictions. AdaBoost also shows significant improvement with the combined features, achieving an MCC of 99.52%. SVM benefits notably, with an MCC of 98.83%, indicating that the combined feature set enhances feature space separation, improving SVM's classification capabilities. LR sees improved performance with an MCC of 97.39%, reflecting better capture of linear relationships with the combined features, though it remains lower than non-linear classifiers. Finally, the DT achieves a consistent MCC of 99.38% with the combined features.

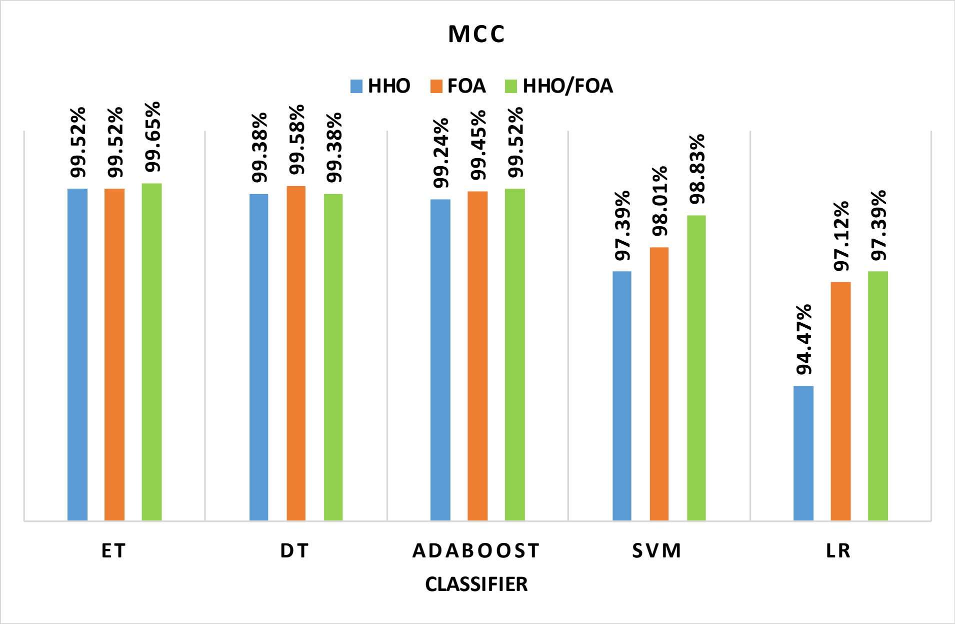


Fig. 9 *MCC of the proposed Spam-FOA-HHO*

In a nutshell, the union of HHO and FOA significantly enhances the model's performance. ET is the most effective classifier because it handles complex interactions and provides balanced predictions. The integration of HHO and FOA for feature selection significantly enhances the performance of multiple classifiers, with the union of both algorithms generally providing the highest performance, thus justifying the effectiveness and robustness of the proposed model in accurately identifying all relevant instances across different classifiers.

6. Conclusion

In this study, we tackled the issue of email spam, proposing the Spam-FOA-HHO model. The Spam-FOA-HHO model employs HHO and FOA to enhance the feature selection process. The Spam ISCX-URL2016 dataset was utilized as a benchmark to test the Spam-FOA-HHO model with several classifiers. Hyperparameters for these classifiers were meticulously optimized using the Grid Search technique to ensure maximum performance. The results reveal that the Spam-FOA-HHO model significantly outperforms the individual applications of FOA and HHO in feature selection. Among the classifiers tested, the ET classifier achieved the highest accuracy of 99.83% with the proposed Spam-FOA-HHO model. Accordingly, the Spam-FOA-HHO model demonstrates superior performance in email spam detection, offering a robust solution that significantly improves upon traditional feature selection methods.

Acknowledgement

The authors would like to thank Al-Ahliyya Amman University for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

*The authors confirm contribution to the paper as follows: **conceptualization**: Mosleh M. Abualhaj; **methodology**: Mosleh M. Abualhaj; **software**: Sumaya Al-Khatib and Mohammad O. Hiari; **validation**: Qusai Y. Shambour; **formal analysis**: Mosleh M. Abualhaj; **investigation**: Qusai Y. Shambour; **resources**: Sumaya Al-Khatib and Mohammad O. Hiari; **data curation**: Sumaya Al-Khatib; **draft preparation**: Mosleh M. Abualhaj and Qusai Y. Shambour; **review and editing**: Qusai Y. Shambour; **visualization**: Qusai Y. Shambour; All authors have read and agreed to the published version of the manuscript.*

References

[1] Maroofi, S., Korczyński, M., Hölzel, A., & Duda, A. (2021). Adoption of Email Anti-Spoofing Schemes: A Large Scale Analysis. IEEE Transactions on Network and Service Management, 18(3), 3184-3196. https://doi.org/10.1109/TNSM.2021.3065422.

[2] Kim, J., Self, J. A., & Park, Y.-W. (2020). Investigating physical interaction with digital data through the materialization of email handling. Interacting with Computers, 32(5-6), 457–474. https://doi.org/10.1093/iwc/iwab003

[3] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. IEEE Access, 7, 168261–168295. https://doi.org/10.1109/access.2019.2954791

[4] Petrosyan, A. (2024, March). Share of spam in email traffic worldwide from 2007 to 2023. Statista. Retrieved May 2024, from https://www.statista.com/statistics/420400/spam-email-traffic-share-annual/.

[5] Anti-Spam Engine. (2024, March). Spam statistics. Retrieved May 2024, from https://antispamengine.com/spam-statistics/.

[6] Makkar, A., Garg, S., Kumar, N., Hossain, M. S., Ghoneim, A., & Alrashoud, M. (2020). An Efficient Spam Detection Technique for IoT Devices using Machine Learning. IEEE Transactions on Industrial Informatics, 1–1. https://doi.org/10.1109/tii.2020.2968927

[7] Abualhaj, M. M., Abu-Shareha, A. A., Hiari, M. O., Alrabanah, Y., Al-Zyoud M., & Alsharaiah, M. A. (2022). A Paradigm for DoS Attack Disclosure using Machine Learning Techniques. International Journal of Advanced Computer Science and Applications, 13(3). https://doi.org/10.14569/ijacsa.2022.0130325

[8] Hani Al-Mimi, Nesreen Adnan Hamad, Abualhaj, M. M., Al-Khatib, S. N., & Hiari, M. O. (2023). Improved Intrusion Detection System to Alleviate Attacks on DNS Service. Journal of Computer Science, 19(12), 1549–1560. https://doi.org/10.3844/jcssp.2023.1549.1560

[9] Almomani, O. (2021). A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System. Computers, Materials & Continua, 68(1), 409–429. https://doi.org/10.32604/cmc.2021.016113

[10] Shambour, Q. Y., Turab, N. M., & Adwan, O. Y. (2021). An Effective e-Commerce Recommender System Based on Trust and Semantic Information. Cybernetics and Information Technologies, 21(1), 103–118. https://doi.org/10.2478/cait-2021-0008

[11] Lingam, G., Rout, R. R., Somayajulu, D. V. L. N., & Ghosh, S. K. (2020). Particle Swarm Optimization on Deep Reinforcement Learning for Detecting Social Spam Bots and Spam-Influential Users in Twitter Network. IEEE Systems Journal, 1–16. https://doi.org/10.1109/jsyst.2020.3034416

[12] Al Saaidah, A., Abualhaj, M. M., Shambour, Q. Y., Ahmad Adel Abu-Shareha, Laith Abualigah, Al-Khatib, S. N., & Alraba'nah, Y. H. (2024). Enhancing malware detection performance: leveraging K-Nearest Neighbors with Firefly Optimization Algorithm. Multimedia tools and applications. https://doi.org/10.1007/s11042-024-18914-5

[13] Al-Mimi, H., Hamad, N. A., & Abualhaj, M. M. (2023, May). A model for the disclosure of probe attacks based on the utilization of machine learning algorithms. In 2023 10th International Conference on Electrical and Electronics Engineering (ICEEE) (pp. 241-247). IEEE.

[14] Sonny, A., Kumar, A., & Linga Reddy Cenkeramaddi. (2023). Carry Object Detection Utilizing mmWave Radar Sensors and Ensemble-Based Extra Tree Classifiers on the Edge Computing Systems. IEEE Sensors Journal, 23(17), 20137–20149. https://doi.org/10.1109/jsen.2023.3295574

[15] Jiang, X., Xu, Y., Ke, W., Zhang, Y., Zhu, Q. X., & He, Y. L. (2022). An imbalanced multifault diagnosis method based on bias weights AdaBoost. IEEE Transactions on Instrumentation and Measurement, 71, 1-8.

[16] Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016). Detecting malicious urls using lexical analysis. In Network and System Security: 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings 10 (pp. 467-482). Springer International Publishing.

[17] Kumar, A., & Soumyadev Maity. (2022). Detecting Malicious URLs using Lexical Analysis and Network Activities. 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA). https://doi.org/10.1109/icirca54612.2022.9985586

[18] Li, J., Wei, X., Li, B., & Zeng, Z. (2022). A survey on firefly algorithms. Neurocomputing, 500, 662–678. https://doi.org/10.1016/j.neucom.2022.05.100

[19] Kumar, V., & Kumar, D. (2020). A Systematic Review on Firefly Algorithm: Past, Present, and Future. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-020-09498-y

[20] Tripathy, B. K., Reddy Maddikunta, P. K., Pham, Q.-V., Gadekallu, T. R., Dev, K., Pandya, S., & ElHalawany, B. M. (2022). Harris Hawk Optimization: A Survey onVariants and Applications. Computational Intelligence and Neuroscience, 2022, 1–20. https://doi.org/10.1155/2022/2218594

[21] Abualhaj, M. M., & Al-Khatib, S. N. (2024). Using decision tree classifier to detect Trojan Horse based on memory data. Telkomnika, 22(2), 393–393. https://doi.org/10.12928/telkomnika.v22i2.25753

[22] Zhou, B., Yao, Y., & Luo, J. (2013). Cost-sensitive three-way email spam filtering. Journal of Intelligent Information Systems, 42(1), 19–45. https://doi.org/10.1007/s10844-013-0254-7

[23] Rathi, M., & Pareek, V. (2013). Spam Mail Detection through Data Mining – A Comparative Performance Analysis. International Journal of Modern Education and Computer Science, 5(12), 31–39. https://doi.org/10.5815/ijmecs.2013.12.05

[24] Zhang, Y., Wang, S., Phillips, P., & Ji, G. (2014). Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowledge-Based Systems, 64, 22–31. https://doi.org/10.1016/j.knosys.2014.03.015

[25] Idris, I., Selamat, A., Thanh Nguyen, N., Omatu, S., Krejcar, O., Kuca, K., & Penhaker, M. (2015). A combined negative selection algorithm–particle swarm optimization for an email spam detection system. Engineering Applications of Artificial Intelligence, 39, 33–44. https://doi.org/10.1016/j.engappai.2014.11.001

[26] Shi, P., Yao, X., He, S., & Cui, B. (2018). Malicious URL detection with feature extraction based on machine learning. International Journal of High Performance Computing and Networking, 12(2), 166. https://doi.org/10.1504/ijhpcn.2018.10015545

[27] Zhao, J., Wang, N., Ma, Q., & Cheng, Z. (2019). Classifying malicious URLs using gated recurrent neural networks. In Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 12th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018) (pp. 385-394). Springer International Publishing.

[28] R, vinayakumar, S, S., KP, S., & Alazab, M. (2020). Malicious URL Detection using Deep Learning. https://doi.org/10.36227/techrxiv.11492622.v1

[29] Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162.

[30] Adil, M., Almaiah, M. A., Omar Alsayed, A., & Almomani, O. (2020). An Anonymous Channel Categorization Scheme of Edge Nodes to Detect Jamming Attacks in Wireless Sensor Networks. Sensors, 20(8), 2311. https://doi.org/10.3390/s20082311

[31] Toghuj, W., & Alraba'nah, Y. (2024). A two-stage approach for aircraft detection with convolutional neural network. International Journal of Electrical and Computer Engineering (IJECE), 14(4), 4627-4635.

[32] Sasi, S., Lilywala, T. Y., & Bhattacharya, B. S. (2022, July). Optimising hyperparameter search in a visual thalamocortical pathway model. In 2022 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[33] Abualhaj, M. M., Ahmad Adel Abu-Shareha, Qusai Shambour, Adeeb Alsaaidah, Al-Khatib, S. N., & Anbar, M. (2024). Customized K-nearest neighbors' algorithm for malware detection. International Journal of Data and Network Science, 8(1), 431–438. https://doi.org/10.5267/j.ijdns.2023.9.012

[34] Munther, A., Abualhaj, M. M., Alalousi, A., & Fadhil, H. A. (2024). A significant features vector for internet traffic classification based on multi-features selection techniques and ranker, voting filters. International Journal of Electrical & Computer Engineering (2088-8708), 14(6).

[35] Alraba'nah, Y., & Toghuj, W. (2024). A deep learning based architecture for malaria parasite detection. Bulletin of Electrical Engineering and Informatics, 13(1), 292-299.