Received 18 August 2022, accepted 29 August 2022, date of publication 5 September 2022, date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204593

RESEARCH ARTICLE

Feature Selection by Multiobjective Optimization: Application to Spam Detection System by Neural Networks and Grasshopper Optimization Algorithm

SANAA A. A. GHALEB[1,2,3,4], **MUMTAZIMAH MOHAMAD**[4], **WAHEED ALI H. M. GHANEM**[1,2,3,5], **ABDULLAH B. NASSER**[6], **(Member, IEEE)**, **MOHAMED GHETAS**[7], **AKIBU MAHMOUD ABDULLAHI**[8], **SAMI ABDULLA MOHSEN SALEH**[9], **HUMAIRA ARSHAD**[10], **ABIODUN ESTHER OMOLARA**[11], **AND OLUDARE ISAAC ABIODUN**[11]

[1]Faculty of Engineering, University of Aden, Aden, Yemen
[2]Faculty of Education Aden, University of Aden, Aden, Yemen
[3]Faculty of Education Saber, University of Lahej, Lahej, Yemen
[4]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu 21300, Malaysia
[5]Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu 32030, Malaysia
[6]School of Technology and Innovation, University of Vaasa, 65200 Vaasa, Finland
[7]Faculty of Computer Science, Nahda University, Beni Suef Governorate 62764, Egypt
[8]Faculty of Computing and Informatics, Albukhary International University, Kedah 05200, Malaysia
[9]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Pulau Pinang 14300, Malaysia
[10]Department of Computer Science, Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[11]Department of Computer Science, University of Abuja, Gwagwalada 900110, Nigeria

Corresponding authors: Sanaa A. A. Ghaleb (sanaaghaleb.sg@gmail.com) and Waheed Ali H. M. Ghanem (waheed.ghanem@gmail.com)

ABSTRACT Networks are strained by spam, which also overloads email servers and blocks mailboxes with unwanted messages and files. Setting the protective level for spam filtering might become even more crucial for email users when malicious steps are taken since they must deal with an increase in the number of valid communications being marked as spam. By finding patterns in email communications, spam detection systems (SDS) have been developed to keep track of spammers and filter email activity. SDS has also enhanced the tool for detecting spam by reducing the rate of false positives and increasing the accuracy of detection. The difficulty with spam classifiers is the abundance of features. The importance of feature selection (FS) comes from its role in directing the feature selection algorithm's search for ways to improve the SDS's classification performance and accuracy. As a means of enhancing the performance of the SDS, we use a wrapper technique in this study that is based on the multi-objective grasshopper optimization algorithm (MOGOA) for feature extraction and the recently revised EGOA algorithm for multilayer perceptron (MLP) training. The suggested system's performance was verified using the SpamBase, SpamAssassin, and UK-2011 datasets. Our research showed that our novel approach outperformed a variety of established practices in the literature by as much as 97.5%, 98.3%, and 96.4% respectively.

INDEX TERMS Spam detection system (SDS), grasshopper optimization algorithm (GOA), feature selection (FS), multi-objective optimization (MOO), multilayer perceptron (MLP).

I. INTRODUCTION

Spam is electronic mail that is not requested yet is sent to large numbers of people. Advertising is usually a known type of spam and is the most common way to send it. But

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

it is incorrect to think that such a strategy is used only in commercial environments. The number of spam e-mails has grown over recent years, whereby a recipient regularly receives heaps of emails on a daily basis, of which 92% are spam [1]. Currently, the battle between spam detection tools and spammers is an ongoing battle as each side seeks new ways to neutralize the other's presence [2].

Ensuring the integrity and privacy of those statistics is turning into an actual challenge. The Simple Mail Transfer Protocol (SMTP) has the ability to transmit and receive emails over the internet, but it does not have security measures built into it. To quote the SMTP [3], the designers of these protocols are aware of the security challenges of SMTP. The SMTP protocol does not include data integrity, encryption, or authentication services [4].

Spam to a private email can cause havoc throughout the system. Nowadays, it has created many problems in business life, such as occupying network bandwidth and the space in users' mailboxes. Research has been conducted in this area to resolve this issue and spam detection systems (SDS) have been developed to monitor spammers and filter email activities by identifying patterns in email messages, thus improving the tool to detect spam [5], [6].

Both the knowledge filtering and the guideline filtering strategies are used to detect spam. Both have advantages and disadvantages, but neither is effective against all threats [7], [8]. The guideline detection method works well for identifying recognised communications but not spam [8]. In comparison, the knowledge detection strategy is effective at finding new messages, but it has a low detection rate and a high percentage of false positives [9]. As such, our study introduces a new method. Most investigations into spam detection in the literature have focused on the knowledge detection strategy since it seemed more promising.

Recently, several methods, including machine learning, statistical analysis, and artificial intelligence techniques, have emerged in the field of knowledge detection [8], [9], [10], [11], [12]. Unsupervised, semi-supervised, and supervised machine learning techniques are the three types used, and in general, supervised learning performs better than the other techniques. Several Machine Learning Algorithms (MLA) can be employed for knowledge identification, including Naive Bayes (NB), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and k-Nearest Neighbor (KNN) [10], [11], [12], [13].

The majority of categorization information is highly dimensional, and for effectiveness and accuracy, natural dimensionality reduction is also required. As a result, the main disadvantage of content classification is its high dimensionality. The features with area addresses will act in conjunction with high dimensionality or an excessive array of options (a large assortment of vocabulary that consists of all the special terms that occur a minimum of once or over once within the collection of emails). Due to the performance of the majority of content classifiers, this drawback worsens the system as a whole. Additionally, it will make the system more complex overall.

Dimensionality reduction is crucially required to handle and combat high spatiality problems as well as mitigate their effects. This work is centred on the dimensionality of unsolicited mail email classifiers.

Thus, the feature selection mechanism may be a curse for the dimensionality of the selection of appropriate features and its classification. However, many features may be lowered, and the training time may increase with the elimination and reduction of redundant features, therby improving the classification's performance. This analysis discussed the several drawbacks of the well-known methods used in earlier feature selection studies. The two types of feature selection algorithms are filters and wrappers. Gain ratio, information gain, chi-squared, and correlation-based feature selection are a few examples of statistical, information theory-based, or searching methods that can be used to apply filters [13], [14]. Wrappers evaluate and categorise capabilities using a machine learning technique to determine the subset that, for the most part, makes up the dataset.

They have been built entirely on the following components: a learning algorithm of a set of rules that may be any classifier, and a feature search, sequential search, genetic search, etc. The wrapper technique often requires less processing than the clear-out strategy, but the latter yields the best results [14].

Some researchers have classified the proposals which are based on artificial intelligence optimization algorithms into the following categories: biology-based, social-based, chemical-based, physics-based, mathematics-based, music-based, sports-based, swarm-based, plant-based, light-based, and water-based [15], [16], [17].

Based on this categorization, our proposal is based on swarms the contributions of this work are summarized as follows:

1. The proposed MOBGOA as a wrapper-based feature selection to determine features from the emails in the first stage.

2. Adapted the EGOAMLPs for the training of supervised Multi-Layer Perceptrons (MLPs) in a second stage.

3. The final SDS approach (MOBEGOAMLP) was tested by three spam datasets (SpamBase, SpamAssassin, and UK-2011 Webspam) on ten statistics.

Section II provides an overview of this study. Section III presents related research. Section IV discusses the methodology. Section V discusses the performance evaluation. The assessment of contributions is depicted in Section VI, alongside results and discussions. The conclusion is in Section VII.

II. BACKGROUND

A. GRASSHOPPER OPTIMIZATION ALGORITHM (GOA)

The GOA was inspired by the behaviour of grasshopper insects and is one of the metaheuristic algorithms that [18] presented in 2017. The grasshopper swarms go through two stages in their life cycle: nymphs and adults. The nymph grasshopper travels slowly over a short distance, which lets them take advantage of their habitat and consume all the vegetation in their way. The adult grasshopper, on the other hand, has two primary responsibilities: locating food and migrating. It has a greater region to explore because it can jump quite high and travel a long way to obtain food. We can infer that the grasshopper's two movements, slow movement over a small distance and abrupt movement over a wide distance, are both indicative of exploitation and exploration. The grasshoppers prefer to move locally during the exploitation stage, whereas during exploration they prefer to wander over long distances in search of food. The accomplishment of these two tasks, as well as locating a food source, is a natural process for grasshoppers. The mathematical model presented in [19], which is replicated here, describes the grasshopper swarming behaviour as follows:

$$X_i = S_i + G_i + A_i \tag{1}$$

where X_i stands for the i^{th} grasshopper's location, S_i for the social interaction, G_i for gravity acting on the i^{th} grasshopper, and A_i for the wind advection. Eq. (1) can be expanded to include S_i, G_i, and A_i, and then rewritten as follows:

$$X_i = \sum\nolimits_{j=1,j\neq i}^{N} s\left(\left|x_j - x_i\right|\right)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + \hat{e_w} \tag{2}$$

where N is the number of grasshoppers and $s(r) = fe^{r/l} - e^{-r}$ is a function that simulates the effects of social interactions. $g\hat{e}_g$ where g is gravitational force and $\hat{e}_g$ is a unit vector pointing toward the center of the earth, is the enlarged G_i component. The extended A_i component is represented as $u\hat{e_w}$, where u is a constant drift and $\hat{e_w}$ is a unit vector heading toward the wind. Where d_{ij} equals $\left|x_i - x_j\right|$ and denotes the separation between the i^{th} and j^{th} grasshoppers. The effects of wind and gravity are much smaller than the relationships between grasshoppers since they discover comfortable zones rapidly and have poor convergence, hence this mathematical model should be changed as follows:

$$X_i^d = c\left(\sum\nolimits_{j=1,j\neq i}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right| \frac{x_j - x_i}{d_{ij}}\right)\right) + \hat{T}_d \tag{3}$$

In Eq. (3) the parameter stands in for the upper and lower ub_d *and* lb_d bounds in the D^{th} dimension, respectively, and $\hat{T}_d$ parameter denotes the best solution value in the D^{th} dimension at the time. As a result, the parameter c must be reduced in accordance with the quantity of iterations. The more iterations there are, the more exploitation is encouraged by this system. The calculation for the argument c, which shrinks the comfort zone according to iterations, is as follows:

$$c = c_{max-Iter\frac{c_{max}-c_{min}}{iter_{max}}} \tag{4}$$

In Eq. (4), the parameters c_{min} and c_{max} stand for the maximum and minimum values, respectively. Iter stands for the most recent iteration and $iter_{max}$ denotes the maximum number of iterations.

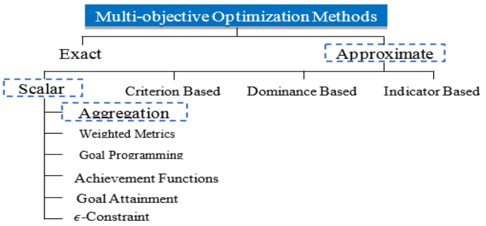


FIGURE 1. Classification of moo algorithms, highlighting the methods used in this research.

B. JUSTIFYING THE GOA ALGORITHM

The inherent benefit of GOA is that it enhances convergence quality by merging single-based and population-based methods. The following are some additional advantages of the GOA that encourage scholars to use it to address classification issues [19]:

- During their initial search, grasshoppers can make a number of abrupt large step hops and can automatically seek into areas where potentially superior solutions have already been discovered.
  ▪ The automatic transition from exploratory movement to local focused exploitation is used to carry out this search. As a result, the GOA converges quickly in the initial phases of the iteration process.
  ▪ The GOA updates the position by taking into account not only the current position of the grasshopper and the position of the target, but also the positions of every other grasshopper.
  ▪ The majority of metaheuristic algorithms use pre-tuned preset parameters. The GOA, in contrast, used parameter control, which involved varying the values of the parameters (*C2* and *C1*) throughout each cycle. This aids in automatically switching the GOA from exploration to exploitation when searching is the optimal course of action.

C. MULTI-OBJECTIVE OPTIMIZATION (MOO)

The MOO is important as it helps make the best decision possible, especially when there are trade-offs between at least two different objective functions. It may involve increasing or decreasing several changing objective functions [20]. The equation for an n-objective minimization challenge's equation is as follows:

$$\text{Minimise} : F(x) = [f1(x), f2(x), f3(x), \ldots, fn(x)] \tag{5}$$

$$\text{Subject to} : g_i(x) \leq 0, \text{i} = 1, 2, 3, \ldots \text{m}, \text{h}_i(x) = 0, \text{i} = 1, 2, 3, \ldots 1 \tag{6}$$

The total number of objective functions that must be lowered, where x is a selection vector, is n. The model in Eq. (6) transforms into a single-objective issue when n is equal to 1, and the perfect solution minimises the objective. On the other hand, when $n > 1$, $f_i(x)$ denotes the objective function, whereas g_i(x) and h$_i(x)$ denote the utility functions of the issue being maximised or minimised.

In MOO, a solution's nature is indicated by the trade-off between its n different aims. The optimum solutions to the MOO problems are all non-dominated arrangements if the following criteria are satisfied: x is dominant over y. The Pareto set/front refers to these solutions [20]:

$$\forall i{:}\, f_i\,(x) \leq f_i\,(y)\; and\; \exists_j{:}f_i\,(x) < f_i\,(y) \tag{7}$$

MOOs are used to gather a collection of trade-offs, drawbacks, or non-dominant options. The Pareto-optimal solution is one that does not outperform any other solution in a given situation.

The Pareto front, a trade-off surface, is defined by all solutions [21]. Scalar methods, criterion-based methodologies, dominance-based methodologies, and indicator-based approaches are the four main groups of MOO metaheuristics. More information is shown in Figure 1 which presents further details [22]. This graphic also shows the MOO approach that we propose.

1) SCALAR APPROACHES

This group of MOO metaheuristics includes approaches that change a MOO problem into a single objective or a collection of similar problems. The strategy shown in Section II is modified to become a scalar methodology. The methodology includes the accumulation strategy, weighted measurements, goal programming, achievement capacities, goal achievement, and ϵ-constraint techniques. Scalarization techniques are used to construct Pareto ideal layouts, which is the justification. The scalar method is an a priori technique; it calls for the communication of sufficient inclination data before the solution procedure. Commonly used examples of priori methods are the utility capacity strategy, goal programming, and lexicographic technique.

2) AGGREGATION METHOD

The aggregation (or weighted aggregation) approach is one of the most important and frequently used methods for producing Pareto optimal solutions. In this approach, an aggregation function is used to connect numerous objective functions f_i linearly into a single objective function f, converting a MOO problem to a single-objective problem.

$$f\,(x) = \sum\nolimits_{i=1}^{n} \omega_i f_i\,(x) \tag{8}$$

where the weights $\omega_i \in [0\ldots 1]$ and $\sum_{i=1}^{n} \omega_i = 1$. The trade-off in FS for SDS comprises the minimisation of classification error rate, the reduction of false alarms, and the quantity of features. FS approaches for SDSs are thus given as a three objective minimization problem. Several approaches are used to optimise the FS process. The evaluation of this particular technique in this research, however, was motivated by the fact that the multi-objective binary GOA algorithm for FS in SDSs has not been studied recently [22].

III. RELATED WORK

There are other related publications in the literature that address different detection strategies. However, knowledge detection remains to be the most popular strategy. Of course, because of its effectiveness in detecting new messages, it is expensive to concentrate on knowledge detection. The hybrid detection strategy has made some progress in recent years [23], [24], but it is still a long way from the knowledge approach and the guideline approach.

Additionally, other SDSs based on knowledge detection have been developed [25] employing a variety of techniques and classifiers. Despite the rise in usage of hybrid classifiers and ensemble classifiers [26], single classifiers are still used and can produce high-quality results. The UK-2011 Webspam dataset, SpamBase, and SpamAssassin are still the three most commonly used datasets for SDS performance evaluation in the literature. Ten measures are the evaluation criteria most papers use to evaluate the performance of their approaches [8].

Instead of using the complete feature space, many spam detection-related research uses a feature selection procedure to choose the best subset of features to represent the whole dataset [27]. The size of the dataset utilised and the classification performance of various algorithms can both be impacted by reducing the feature space [25]. This can be accomplished using a variety of techniques. Even though it takes more time and computer resources, the wrapper approach for feature selection performs better than the alternatives [28]. Intriguingly, email spam detection has recently seen a large increase in the adoption of natural inspired methodologies.

With the help of Bayesian theory as a fitness function and checked with a different number of repeats, a new hybrid SDS is proposed in [9] that uses a GA algorithm to select features without a fixed number of feature selections. From the 57 features in the SpamBase dataset, they only obtained 38. The examples are finally classified using a Naive Bayes classifier technique on the smaller data set. However, a large number of characteristics contributed to a high-dimensional space.

Using particle swarm optimization (PSO) and correlation-based feature selection (CFS), [11] proposed a novel hybrid SDS. The CFS-PSO leads the method to create a logical model with enhanced performance. From the 24 features in the UK 2006 dataset, they only managed to extract 6 features. The instances are finally classified using an MLP and NB classifier technique in the smaller data set, which results in classification AUCs of 16.13% and 8.23%.

In [27] proposed a new hybrid SDS that incorporates the Water Cycle and Simulated Annealing (WCSA). The WCSA is used to remove redundant and unnecessary features that could obstruct performance. The instances are finally classified using an SVM classifier technique in the smaller data set. From the 57 features in the SpamBase dataset, they extracted 26 features.

The case is finally classified using a KNN classifier, which yields a classification accuracy of 94% for the smaller dataset. As part of the algorithm, WOA develops solutions in their search space [25] using the prey siege and encirclement process, bubble invasion, and search for prey methods in an effort to improve the FS problem's solutions. In addition, FPA enhances the FS problem's solutions using two global and local search processes in a search space that is opposite from the solutions of WOA. In actuality, they employed every potential answer to the FS problem from both the solution search space and its opposite. Experiments were run in two steps to assess the performance of the suggested method. Ten FS datasets from the UCI data repository were used for the tests in the first stage.

A new hybrid SDS using the Binary Firefly Algorithm (BFA) was proposed in [29]. The choice of a feature is based on a fitness function that is reliant on the acquired accuracy when using a Naive Bayesian Classifier (NBC), and BFA explores the space of the best feature subsets. The FA approach has a sluggish convergence rate and requires expensive computing. Of the 57 features in the Spambase dataset, 21 features were extracted. The examples are finally classified using an NBC algorithm in the smaller dataset, which yields a classification accuracy of 95.14%.

Using a Genetic Algorithm (GA) and Random Weight Network (RWN), [30] suggested a novel hybrid SDS. Using RWN GA determines the optimum feature subsets based on the accuracy it has been able to accomplish. In spite of this, GA uses a lot of resources. From the 57 features in the SpamAssassin dataset, they only isolated 25 features. The examples are finally classified using an RWN classifiers algorithm in the smaller dataset, which yields a 92% classification accuracy.

In [31], proposed a novel spam classification technique using Naive Bayes (NB) and Support Vector Machines (SVM). They obtained 80 of the 140 features contained in the Spamassassin dataset. Finally, the SVM/NB classifiers are used in the reduced dataset to classify the instances and achieve a classification accuracy of 97%, and 98%.

In [32] a web spam detection method by extracting novel feature sets from the homepage source code and choosing the random forest (RF) as the classifier against the UK-2011 dataset was proposed. Finally, an RF classifier is used in the reduced dataset to classify the instances and achieve a classification accuracy of 93%.

To get over the problem of false drift, [33] presented a Disposition Based Drift Detection Method (DBDDM), a DBDDM. In order to determine the actual drift, this study uses the approximation randomization test to calculate the frequency of successive drift and compares the frequency with the threshold. When Naive Bayes (NB) and the Hoeffding tree (HT) classifier are used, it shows a maximum gain in accuracy of 24% and 28% and an increase of 2.50 and 1.91 average ranks, respectively.

A novel hybrid SDS based on PSO and Fruit Fly Optimization (FFO) based on PSO for Feature Selection was proposed as a novel hybrid SDS in [34]. An FFO is utilised to optimise the PSO. These methods do not, however, perform as well in local and international searches. From the 57 features in the SpamBase dataset, they only extracted 10 features. The cases are finally classified using an FFOPSO classifier algorithm in the smaller data set.

TABLE 1. Comparisons of related work.

Work	FS components	No FS	Classifier	Results	Dataset
[35]	HHO	57	KNN	94.3%	SB
[31]	-	80	SVM/NB	97/98%	SA
[32]	-	-	RF	93%	UK
[29]	BFA	21	NBC	95.14%	SB
[25]	HWOAFPA	48	KNN	94%	SB
[27]	WCSA	26	SVM	96.3%	SB
[9]	GA	38	NB	94.5%	SB
[34]	FFOPSO	10	KNN	94.82%	SB
[33]	DBDDM	-	HT/NB	91.34/91 .64%	SA
[30]	GA	25	RWN	92%	SA
[11]	CFS-PSO	10	MLP/NB	16.13/ 8.23%	UK

Feature Selection ➔FS, SpamBase➔ SB, SpamAssassin➔SA
UK2011Webspam➔ UK, Accuracy➔ ACC

In [35] proposed a new SDS that incorporates the Harris Hawks Optimizer (HHO). The HHO is used to remove redundant and unnecessary features that could obstruct performance. The instances are finally classified using a KNN classifier technique in the smaller data set. A summary of the related work is given in Table 1.

In light of the MOBGOA algorithm, this study provided an SDS model using a different metaheuristic, MOBGOA, with the ultimate goal of MOO FS. The wrapper method of FS is used in this strategy, and the most promising updated GOA model is used to train the MLP because it is acclimated to dealing with tackling the problems MLPs face.

IV. METHODOLOGY THE STUDY

The present methods have performed well in terms of addressing the SD issue. The ideal device, however, has yet to be developed, as it must be able to detect all messages without creating a fake alert in order to provide complete protection from spam. Researchers must contend with a number of obstacles, such as the constant growth of hacking tools, the vast array of existing and emerging data mining and machine learning approaches, the high dimensionality of datasets, etc.

A function selection approach within the framework of SD is laid out in this section. Wrappers outperform filters and deliver better results, but use more processing resources [36]. For that reason, we used MOBGOA as the wrapper method to carry out the function selection. The most efficient metaheuristic can handle this challenge since the range of functions is crucial [37]. Three phases: preprocessing, feature selection, and classification, are integrated into the implementation of the suggested SDS solution. Figure 2 presents the system SDS suggested.

A. PREPROCESSING PHASE

Different types of characteristics, including symbols and characters, are present in the dataset employed in the context of spam identification [38], [39]. The normalisation Eq. 9. is used to normalise these numerical values. A feature extraction tool is used to transform the raw email formats into numerical values, as is the case

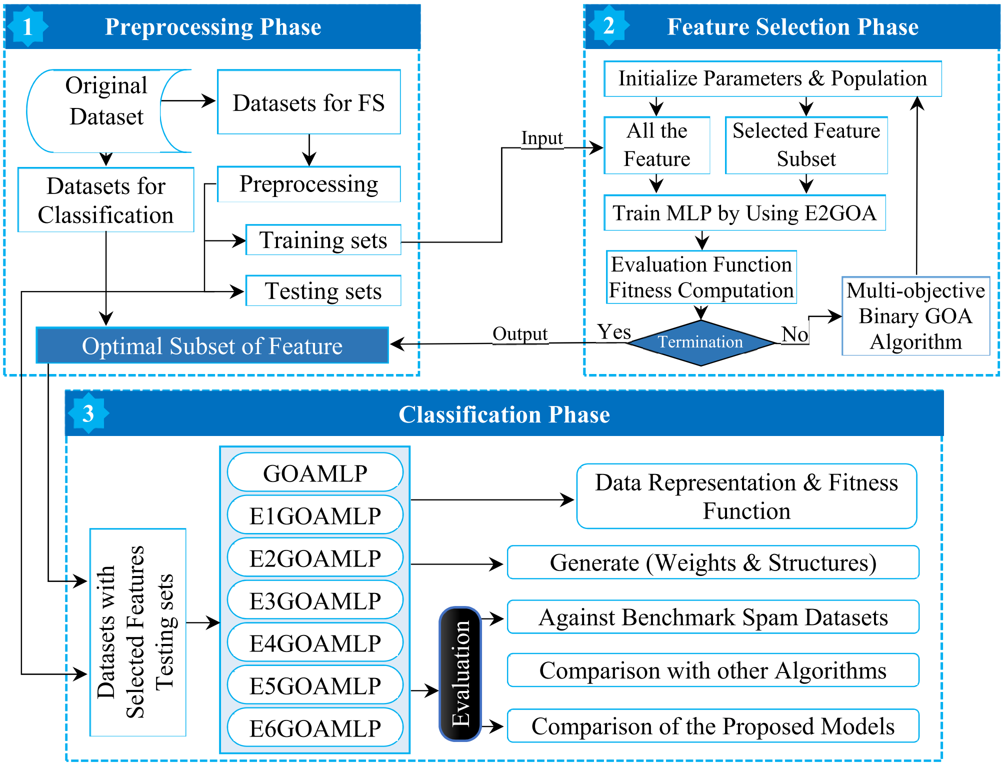

1 Preprocessing Phase
Original Dataset
Datasets for FS
Datasets for Classification
Preprocessing
Training sets
Testing sets
Optimal Subset of Feature
Input
Output
2 Feature Selection Phase
Initialize Parameters & Population
All the Feature
Selected Feature Subset
Train MLP by Using E2GOA
Evaluation Function Fitness Computation
Yes
Termination
No
Multi-objective Binary GOA Algorithm
3 Classification Phase
Datasets with Selected Features Testing sets
GOAMLP
E1GOAMLP
E2GOAMLP
E3GOAMLP
E4GOAMLP
E5GOAMLP
E6GOAMLP
Evaluation
Data Representation & Fitness Function
Generate (Weights & Structures)
Against Benchmark Spam Datasets
Comparison with other Algorithms
Comparison of the Proposed Models

FIGURE 2. System architecture for the proposed SDS.

with the Spam Assassin data set, and it may be found at this URL:"https://github.com/7ossam81/EmailFeaturesExtraction" [40]. Normalization's primary goal is to bring the numerical values of various attributes into the same range. Before using the dataset in the training and testing phases, all of the dataset's characteristics must be normalized. In the datasets, the feature values are meant to give regular semantics. Through the use of Eq. (9), the values are transformed into the range [0, 1], putting all features on the same scale.

$$x_{new} = \frac{x_{current} - x_{min}}{x_{max} - x_{min}} \tag{9}$$

Each collection of features in each of the 3 datasets utilised in this work has a class, which is either not spam or spam email. As a result, each entry in the dataset falls into either the non-spam or spam category. Each class's value is assigned a numeric value, with the non-spam email class being assigned the No. 0 and the spam email class being assigned the number 1. Preprocessing the entire dataset takes time since it is large to load into memory. Records from the dataset are chosen at random as samples. Then two subsets of this random sample are created; the first is referred to as the training and testing dataset.

B. THE FEATURE SELECTION PHASE

1) DESIGN OF MULTI-OBJECTIVE BINARY GRASSHOPPER OPTIMIZATION ALGORITHM (MOBEGOA)

The most important issue to consider when developing a reliable approach for spam detection systems (SDS) is to focus on two stages for functions are: 1) selecting important features and excluding unimportant features from email data; and 2) developing an approach with a high potential for detecting spam email. The general concept used in this study of the normal feature algorithm selection, which is divided into five basic steps, The first step begins with initialising the original feature set found in all three datasets.

The dimensionality of the search space (SS) frequently affects the initialization method for the MOO binary GOA algorithm. It is important to note that in this paradigm, features are frequently defined as the total number of all possible features. The first step of the protocol corresponds to the initialization phase of the MOBGOA. The candidate features are then discovered in the second step. It's a method of discovery that starts with the creation of a random subset of features that MOBGOA has identified as potential solutions.

The third step is an evaluation procedure of the candidate features. It is an evaluation procedure that begins with using the E2GOAMLP algorithm to train multi-layered neural networks. The E2GOAMLP algorithm is better understood by the interested reader compared to earlier research [41]. The feature selection process is one of the most crucial processes, and the feature selection algorithm is based on a wrapper algorithm. This step is critical in directing the algorithm's selection of an optimal subset of attributes.

In the fourth step, a conditioning procedure to determine the relevant subset or optimal feature subset. It is a conditioning procedure that begins with determining whether to continue or stop the search for other subsets of features by testing the stop criterion. Here, the stop criterion depends on either reaching the maximum No. of predefined iterations or a predefined No. of selected features.

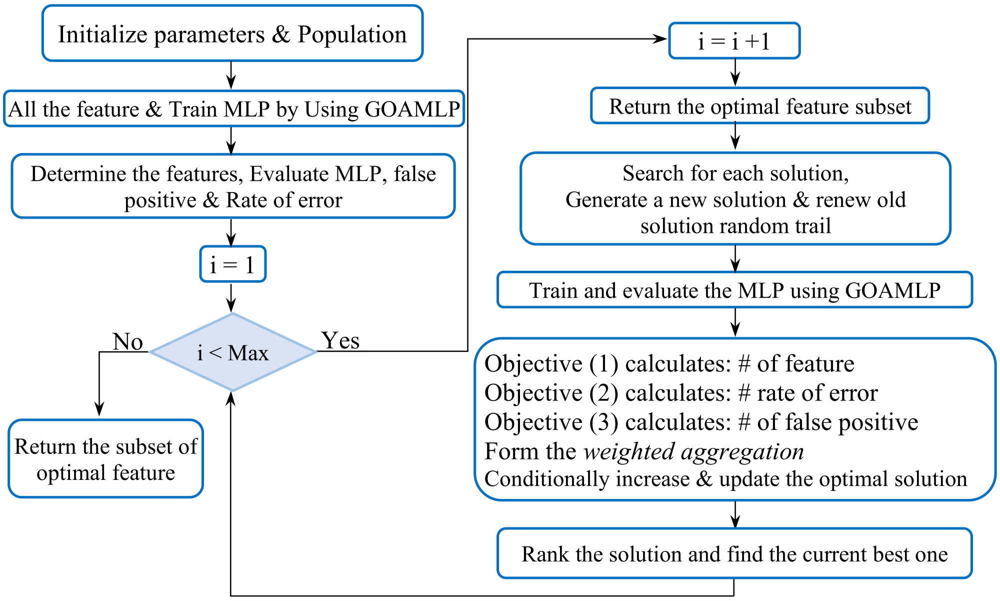

Initialize parameters & Population
All the feature & Train MLP by Using GOAMLP
Determine the features, Evaluate MLP, false positive & Rate of error
i = 1
i < Max
No
Yes
Return the subset of optimal feature
i = i +1
Return the optimal feature subset
Search for each solution, Generate a new solution & renew old solution random trail
Train and evaluate the MLP using GOAMLP
Objective (1) calculates: # of feature
Objective (2) calculates: # rate of error
Objective (3) calculates: # of false positive
Form the *weighted aggregation*
Conditionally increase & update the optimal solution
Rank the solution and find the current best one

FIGURE 3. MOBGOA workflow.

In the fifth step, a result validation of the candidate features is performed. It is a discovery procedure that begins with validating against the 3 datasets. The findings of this phase will be reviewed with those of earlier phases. The MOBGOA algorithm is illustrated in Figure 3, which shows the essential processes in Figure 5, while the following subsections provide additional detail on the method's main components.

2) WRAPPER FEATURE SELECTION METHOD USING EGOAMLP

MOBGOA is employed as a wrapper-based feature selection algorithm. As a result, a wrapper classifier is required for the MOBGOA algorithm to evaluate the subsets. In other words, the Section IV presented MOBGOA algorithm is the multi-objective binary feature selector and EGOAMLP-based evaluator wrapper classifier. FIGURE 3 shows the important role played by the EGOAMLPs algorithm in the bottom loop of the workflow.

With each new generation, the MOBGOA algorithm generates new solutions (a new subset of features is generated). It is entered into the MLP that is trained by the best enhanced GOA (which is introduced in our previous work [41]). The method is employed by employing the novel feature set, and feedback on the method is obtained from the performance of the E2GOAMLP algorithm, which calculates the three objectives and arranges a new solution.

3) MOBGOA PARAMETERS

The MOBGOA algorithm utilises similar parameters to the first GOA model. *C1*, *C2*, and the maximum number of generations to hunt for solutions are the control parameters in GOA. In this study, the maximum number of generations was 1000 and the population size of NP is 50. The MOBGOA algorithm is run 100 times, and the generations in each experiment are terminated upon reaching the maximum number. The number of characteristics to evaluate the final strategy determines the size of the solution space for each dataset used in the study.

4) BINARY ENCODING

The representation and formatting of data is a crucial step before processing data using any ML technique. In the majority of ML classification algorithms, a good representation model is crucially important. In this study, the feature-value representation system was investigated. In keeping with this, every instance in this framework is shown as a vector for characterising the problem domain. The network traffic is set aside as a dataset that is typically handled as a table, with each row addressing a particular occurrence and each column addressing a different network element.

In the MOBGOA, a solution is represented by an n-bit string, where n is the total No. of features in the dataset. The solution's (x_d) value at the d^{th} place is in the range [0, 1], showing the likelihood that the d^{th} feature will be selected. Using the threshold is an additional strategy. A threshold (θ) is used to determine whether or not a feature is selected. If ($x_d > \theta$), the d^{th} feature is enabled; otherwise, it is not. Thus, the normal features are used to create the new subfeatures. MOBGOA employs the threshold strategy. A novel feature in Figure 4 that can be seen as a potential solution is a subset that is uniquely recognised by a binary string.

5) MULTI-OBJECTIVE OPTIMIZATION (MOO)

The concept of MOO of the MOBGOA model is the main feature. The coordination of binary strings on multiple objectives to evaluate solutions in feature selection (FS) problems, rather than visualising on one criterion as accuracy. If the needed solution is a minimization problem, that is, the minimum value of the fitness role, the result is best, and vice versa for maximisation problems. If many goals that require a corresponding fitness function are found, there is a potential

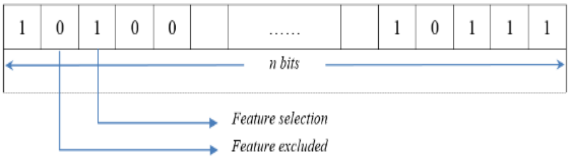


FIGURE 4. Representation of a possible solution as binary string.

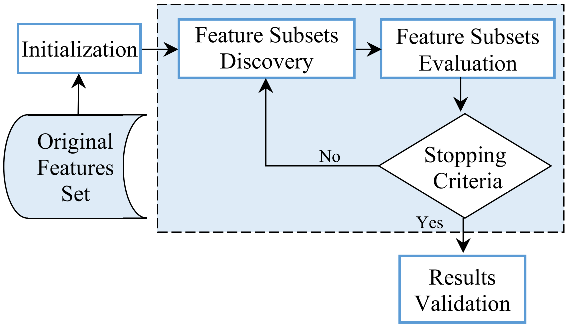


FIGURE 5. The general feature selection process.

conflict between their judgement about the quality of the same solution.

It is worth mentioning that the three objectives with their desirable characteristics are described above:

- FS → to be minimum
- ER → to be minimum
- FPR → to be minimum

The weighted aggregation objective (WAO) that MOBGOA uses to determine performance subsets of feature sets of MLP ratings is illustrated:

$$(WAO) = w_1 \times FS + w_2 \times ER + w_3 \times FPR \qquad (10)$$

From Eq. (10), where w_1 refers to the feature weight and w_2 refers to the error weight, then w_3 refers to the stand for false positive weight. Furthermore, the weights w_2 and w_3 refer to more than $w_{1.}$ In addition, the number of selected features (FS) is no more important than the false positive rate (FPR) of error rate (ER). The three weights (w_1, $w_{2,}$ and w_3) values in the evaluated tests are as follows (0.1, 0.5, and 0.4).

6) COMPUTATIONAL COMPLEXITY

The number of solutions, also known as the D, and the No. of populations, also known as the population size, of the MOBGOA algorithm, are what essentially define the computing complexity of the enhanced GOA method.

The total computing complexity, in the worst case, is O (DNP) ≈ O (O (calculate the GOA position of all solutions and evaluate its fitness) + O (sort solutions of population and GOA population)).

The MOBGOA algorithm's generative process analyses the time complexity of the generation as follows:

The starting population's creation is the primary activity in stage 1, and the time complexity is O (NPD).

Stage 2 temporal complexity for decision-making based on stop/termination criteria is O (1).

Stage 3 involves calculating the value of an aggregated objective parameter based on 3 objectives, namely the No. of features (NF), the error rate (ER), and the false positives (FP), time complexity is O (1). The time complexity in Stage 4, updating the answer, is O (N). Generating continues in Stage 5 and returns back to Phase 2. Consequently, the MOBGOA algorithm's time complexity is O. (NPD).

7) INTEGRATING MOBGOA WITH E2GOAMLP FOR SPAM DETECTION

This design is a spam detection strategy based on EGOA-trained MLP and a set of optimised features. There are two primary components to this goal: Feature selection is the first stage, and classification is the second stage.

The MOBGOA method handles the feature selection portion, while the MLP trained with the E2GOAMLP algorithm handles the classification. Figure 6 depicts how each of these components fits into the overall spam detection image.

It is worth mentioning that E2GOAMLP is also utilised as a wrapper classifier for feature selection with the MOBGOA box in the diagram. The following selection of the best features utilises the MLP trained by E2GOAMLP as the wrapper classifier based on the characteristics. After extracting the features through MOBGOA, the performance of the extracted features is tested, as well as the spam model, termed MOBE2GOAMLP, are both tested using this unified model in the following experimental assessments.

C. THE CLASSIFICATION PHASE

The parameter initialization, data input, ANN training, and EGOA module are the four basic phases of the model. The EGOA system and the ANN model's parameters are initialised during the phase before. The Population Size (NP) parameter, which deals with the population's total number of solutions, is one of many variables in the EGOA algorithm. Each answer (I = 1, 2,..., D) deals with a D-dimensional vector, where D is the total number of elements that influence a decision.

The best solution vectors found up to this point are organised into a grid called Solution Memory (SM). It is an expanded NP-by-D matrix. Before starting the operation, the FS size is modified. In light of the objective function f(x), each solution vector is additionally coupled with a positive value. The algorithm is shown in Figure 7.

The data entry stage is the crucial part of data input in the following step. It relies on how the raw data is transformed, filtered, and how the features are extracted. The split of the raw data into the training and testing sets is a crucial stage. The following component uses it as input information. The approaching data sources should fit into the range of 0 to 1 before the data is fed into the ANN model. This normalisation technique is important for the training in the following module.

The third stage is when the MLP model begins to function after receiving training features for the input data measurement from the information input components. This part is designed as an MLP, or organisation, using Feed-Forward Neural Networks (FFNN). The three-layered neurons that make up the MLP's design are divided into an info layer, a concealed layer, and a yield layer. The MLP module

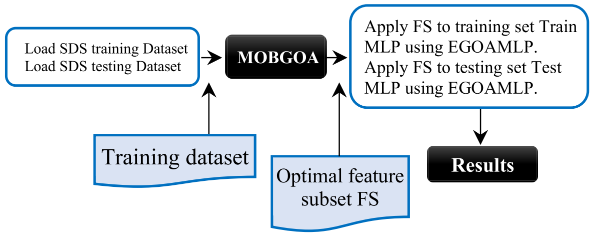

Load SDS training Dataset
Load SDS testing Dataset
MOBGOA
Apply FS to training set Train MLP using EGOAMLP.
Apply FS to testing set Test MLP using EGOAMLP.
Training dataset
Optimal feature subset FS
Results

FIGURE 6. Integrating mobgoa with EGOAMLP.

receives the information from the information input module that is regarded as the designing information (designing dataset) for designing the MLP. It is noteworthy that the EGOA component receives the loads and inclinations in order to carry out the preparation interaction in this module.

The EGOA module is used in the fourth stage as a standalone framework (Black Box) to create novel arrangements that rely on the periodic refreshing of synaptic loads and inclinations. The EGOA module delivers each arrangement as a collection of loads and predispositions into the MLP component during each cycle of the preparation interaction. In this way, each preparation dataset-dependent arrangement is evaluated, and then its wellness values are restored. In this work, the Mean Square Error (MSE) and Fitness Function (FF) are used to process wellbeing. By reducing the MSE estimation of the mistake rate, the loads and inclinations are acquired.

Once the maximal number of cycles is reached, the preparation interaction ends. The loads and predispositions knowledge base is then updated. The EGOA algorithm is linked to other systems for streamlining. As a result, the goal is understood as either increasing or decreasing a measure achieved through this FF. The goal of such a FF should be similar to its value in enhancing calculations. Other than that, its objective is to reduce general error, similar to studying methods demonstrated by previous exams [42], [43]. Therefore, the FF stated before might apply any of the MLP error estimation equations or derive another wellness metric from the recipes. MSE is used in this work as the primary quality component of the proposed EGOA preparation calculation. The preparation goal is to, at its most basic, restrict the MSE to arriving at the highest aggregate of emphasis.

The best classification, approximation, or prediction accuracy for training and testing samples is the main goal of training the MLP. Figure 7 shows the forward pass calculation measure. The fitness function was calculated in this work using a methodology that has been employed in a number of studies [42], [43]. The output of the i^{th} hidden node is determined as follows: If the number of input nodes is N, the number of hidden nodes is H, and the number of output nodes is O.

$$f\left(S_j\right) = Sigmoid\left(S_j\right) = 1\Big/\left(1 + exp\left(-\left(\sum\nolimits_{i=1}^{N} \mathcal{W}_{ij}.\mathcal{X}_i - \beta_j\right)\right)\right), \quad j = ,2, \ldots, H \tag{11}$$

$\mathcal{W}_{ij}$ is the connection weight from the i^{th} node in the input layer to the j^{th} node in the hidden layer, $\mathcal{X}_i$ is the i^{th} input and β_j is the bias (threshold) of the j^{th} hidden node. Where $S_j = \sum_{i=1}^{N} \mathcal{W}_{ij}.\mathcal{X}_i - \beta_j$. The final output can be described as follows after computing the hidden nodes' outputs:

$$O_k = \sum\nolimits_{i=1}^{N} \mathcal{W}_{kj}.f\left(S_j\right) - \beta_k, k = 1, 2, \ldots, O, \tag{12}$$

where β_k is the bias (threshold) of the k^{th} output node and $\mathcal{W}_{kj}$ is the connection weight from the i^{th} hidden node to the k^{th} output node. The following are the calculations made to determine the learning error E (fitness function).

$$E_k = \sum\nolimits_{i=1}^{O} \left(O_i^k - d_i^k\right)^2 \tag{13}$$

$$\mathrm{MSE} = \sum\nolimits_{\mathrm{k}=1}^{\mathrm{q}} \frac{\mathrm{E_k}}{\mathrm{q}} \tag{14}$$

d_i^k is the desired output of the i^{th} input unit when the k^{th} training sample is used, and O_i^k is the actual output of the i^{th} input unit when the k^{th} training sample is used. Where q is the number of training samples, consequently, the following definition applies to the fitness function of the i^{th} training sample:

$$\mathrm{Fitness}(x_i) = \mathrm{MSE}(x_i) \tag{15}$$

V. PERFORMANCE EVALUATION

A. SPAM DATASETS

The evaluation of the proposed ANN system for the specific purpose of SDS defines the usage of benchmark datasets for this particular framework, unlike the datasets used for classification. In this section, three datasets that can be used to test SDSs are briefly explained.

1) SPAMBASE DATASET

In 1999, Hopkins provided the SpamBase dataset [44]. Several writers have utilised this dataset for categorization. This dataset included 4601 emails with an average of 57 attributes, of which 1813 (39%) were spam and 2788 (61%) were not. The dataset's features are all displayed in Table 2.

The percentage of times the special characters “;”,“(“,”[“, “!,” “$,” and “#” appear among the remaining six features is unknown. The other three elements serve as a visual depiction of various capitalization measures used in the messages' text. Finally, each instance's class label can be either 0 for non-spam or 1 for spam. The SpamBase dataset is one of the best for learning and assessment methodologies.

2) SPAMASSASSIN DATASET

The most well-known and often used dataset for identifying spam is the SpamAssassin dataset, which Justin Mason created in 2002 [45]. Information about this dataset can be found at (https://wiki.apache.org/spamassassin) contains information on it. In the 6047 communications that comprised this dataset, there were 1897 unsolicited (spam) emails

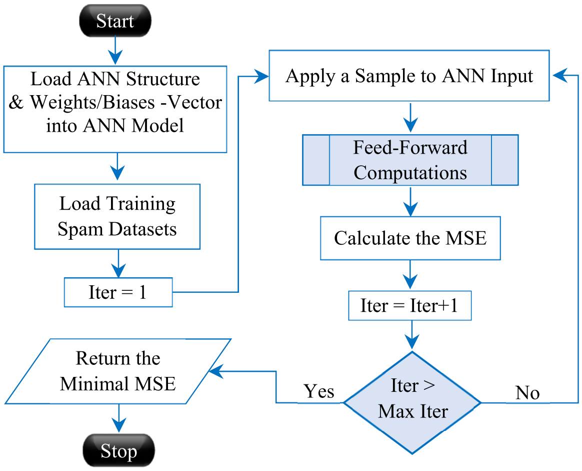

Start
Load ANN Structure & Weights/Biases -Vector into ANN Model
Load Training Spam Datasets
Iter = 1
Apply a Sample to ANN Input
Feed-Forward Computations
Calculate the MSE
Iter = Iter+1
Iter > Max Iter
Yes
No
Return the Minimal MSE
Stop

FIGURE 7. The egoamlp training algorithm flowchart.

(31.4%), 3,900 easy ham emails, and 250 difficult but genuine emails that in many ways resemble spam. In Table 3, the following characteristics of SpamAssassin email messages are displayed:

3) UK-2011 WEBSPAM DATASET

The UK-2011 Webspam dataset consists of 3,766 Web pages with 11 features, 1768 of which are non-spam, and 1998 of which are spam (53%) of the emails, making the data unbalanced and hence more difficult. All of the dataset's features are listed in Table 4, and a detailed description of each feature can be found in [46] and [47].

B. EVALUATION METRICS

Utilizing the accompanying metrics for ACC, FAR, DR, specificity, sensitivity, F-measure, Matthews correlation coefficient (MCC), and G-mean (GM), the effectiveness of the proposed technique is evaluated. The true positives TP, true negatives TN, false positives FP, and false negatives FN cases are used to determine the FAR, DR, MCC, GM, and ACC.

The confusion matrix for a two-class classification in Table 5 yielded these four key criteria. Some performance indicators are used to describe the confusion matrix in Table 6. The performance metrics given in Equations (16–25) are shown in Table 7.

VI. EVALUATION OF MOBEGOAMLP

The suggested MOBGOA framework is thoroughly evaluated in relation to MOBEGOAMLP, thereby confirming the execution of the subsequent SDS method. The three datasets given in Section B are used to test the approach.

A. SPAMBASE RESULTS

In this Scenario 1, the MOBGOA was first applied to the SpamBase dataset to select suitable features from the dimensionality of the search space using the fitness function, resulting in 57 to 15 features as shown in Figure 2. The classification of the resulting features for training and the results obtained are presented in Figure 8. Classification results are displayed using the selected features extracted by MOBGOA training for each training set as presented in Table 9. The proposed MOBE2GOAMLP algorithm is highlighted in bold text.

TABLE 2. All features of the spambase dataset.

No. of Feature	Feature type	Feature description
48	word_freq_WORD	Word frequency expressed as a percentage
6	char_freq_CHAR	Char frequency expressed as a percentage
1	capital_run_length_average	Average length of uninterrupted sequences of capital letters
1	capital_run_length_longest	Average length of uninterrupted sequences of capital letters
1	capital_run_length_total	Total number of capital letters in the e-mail
1	Class feature	Class feature {0 = Not spam, 1 = Spam}

TABLE 3. All features of the spamassassin email messages.

Class	No. of Message	Illustrate
Spam	500	Received from non-spam trap sources
Easy_Ham	2500	Non-spam messages. Easy to differentiate from spam. Do not contain any spam signature (E.g., HTML)
Easy_Ham_2	1400	Non-spam messages
Spam_2	1397	Spam Messages
Overall	5797	33% Spam ratio

TABLE 4. Analysis of UK-2011 WEBSPAM email messages.

No. of Features	Feature	No. of Features	Feature
1	amount of anchor text	7	min_length
2	meta_char	8	average_length
3	meta_word	9	title_words
4	unique_word	10	comp_ratio
5	total_word	11	img
6	max_length		

As per Figure 9, the same results are described in a confusion matrix. Using the definitions in Section B, the experimental results of the suggested EGOAMLPs models are calculated in Table 8. The spam detection model EGOAMLP is able to achieve the very best ratios across the three criteria: DR records of 98.1%, ACC records of 97.5 %, and FAR records of 0.033, according to the acquired results, which were carried out utilising 15 features.

Figure 8 illustrates the convergence curve resulting from sample runs of the GOAMLP, E1GOAMLP, E2GOAMLP, E3GOAMLP, E4GOAMLP, E5GOAMLP, and E6GOAMLP algorithms against selected results from the SpamBase dataset.

Figure 9 shows that the MOBGOA algorithm enhanced the classification accuracy by selecting a subset of 15 features. All the results in the matrices match those listed in Table 8. Due to the constrained space, Figure 9 presents the revised

TABLE 5. The confusion matrix for classification.

Predicted \ Actual	Spam	Non-Spam	Total
Spam	TP	FP	TP+FP
Non-Spam	FN	TN	FN+TN
Total	TP+FN	FP+TN	

TABLE 6. Performance indicators used to describe the confusion matrix.

Type	Definition
TP	shows that an e-mail categorised as unsolicited mail is truly unsolicited mail
TN	shows that an e-mail classified as normal e-mail is sincerely non-unsolicited mail.
FP	Represents a normal mail this is categorized as a spam e-mail.
FN	Represents a spam e-mail this is categorized as a normal mail.

TABLE 7. Mathematical formulae of performance metrics.

Measure	Classification	
Accuracy (ACC)	$ACC = \dfrac{TP+TN}{TP+TN+FP+FN}$	(16)
False alarm rate (FAR)	$FAR = \dfrac{FP}{FP+TN}$	(17)
Detection rate (DR)	$DR = \dfrac{TP}{TP+FN}$	(18)
Sensitivity (SN)	$SN = \dfrac{TP}{TP+FN}$	(19)
Specificity (SP)	$SP = \dfrac{TN}{TN+FP}$	(20)
Positive predictive value (PPV)	$PPV = \frac{TP}{TP+FP}$	(21)
Negative predictive value (NPV)	$NPV = \dfrac{TN}{TN+FN}$	(22)
F-Measure (F1)	$F1 = \dfrac{2 \times PPV \times SN}{PPV+SN}$	(23)
Matthews correlation coefficient (MCC)	$MCC = \dfrac{(TP \times TN - FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN}}$	(24)
G-mean (GM)	$GM = \sqrt{SN \times SP}$	(25)

TABLE 8. The classification results after using algorithms against selected subsets testing of the SPAMBASE.

Alg.	ACC	DR	FAR	MCC	PPV	NPV	SN	SP	F1	G
GOAMLP	92.4	89.8	0.037	0.85	0.97	0.86	0.90	0.96	0.93	93.0
E1GOAMLP	94.5	94.0	0.048	0.89	0.97	0.91	0.94	0.95	0.95	94.6
E2GOAMLP	**97.5**	**98.1**	**0.033**	**0.95**	**0.98**	**0.97**	**0.98**	**0.97**	**0.98**	**97.4**
E3GOAMLP	94.7	94.3	0.046	0.89	0.97	0.92	0.94	0.95	0.96	94.8
E4GOAMLP	93.0	92.8	0.068	0.85	0.95	0.89	0.93	0.93	0.94	93.0
E5GOAMLP	93.4	93.3	0.064	0.86	0.96	0.90	0.93	0.94	0.94	93.4
E6GOAMLP	93.7	92.8	0.050	0.87	0.97	0.90	0.93	0.95	0.95	93.9

proposed model MOBE2GOAMLP with confusion matrices. It's important to highlight that their selection was arbitrary.

TABLE 9. The classification results after using algorithms against selected subsets testing of the SpamAssassin.

Alg.	ACC	DR	FAR	MCC	PPV	NPV	SN	SP	F1	G
GOAMLP	93.9	93.1	0.044	0.87	0.98	0.87	0.93	0.96	0.95	94.3
E1GOAMLP	94.0	93.6	0.053	0.87	0.97	0.88	0.94	0.95	0.95	94.2
E2GOAMLP	**98.3**	**98.3**	**0.018**	**0.96**	**0.99**	**0.97**	**0.98**	**0.98**	**0.99**	**98.3**
E3GOAMLP	95.6	95.1	0.035	0.90	0.98	0.91	0.95	0.96	0.97	95.8
E4GOAMLP	95.6	95.8	0.047	0.90	0.98	0.92	0.96	0.95	0.97	95.5
E5GOAMLP	96.4	96.4	0.037	0.92	0.98	0.93	0.96	0.96	0.97	96.4
E6GOAMLP	91.1	91.5	0.097	0.80	0.95	0.84	0.92	0.90	0.93	90.9

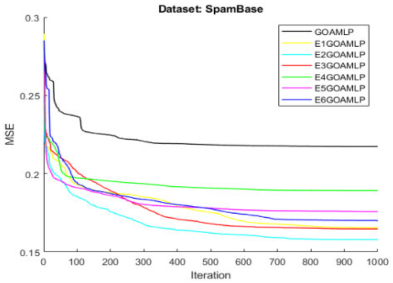


FIGURE 8. Convergence curve of proposed for training the spambase dataset.

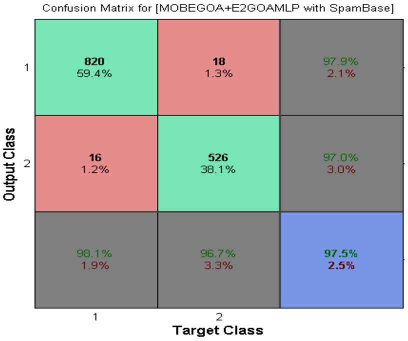


FIGURE 9. Confusion matrices for mobe2goamlp against the spambase dataset.

B. SPAMASSASSIN RESULTS

In this scenario 2, the MOBGOA was first applied to the SpamAssassin dataset to select suitable features from the dimensionality of the search space using the fitness function, resulting in 140 to 48 features as shown in Figure 2. The classification of the resulting features for training and the results obtained are presented in Figure 10. Classification results are displayed using the selected features extracted by MOBGOA training for each training set as presented in Table 9. The proposed MOBE2GOAMLP algorithm is highlighted in bold text. As per Figure 11, the same results are described in a confusion matrix. The spam detection model EGOAMLP is able to achieve the very best ratios across the three criteria: DR records of 98.3%, ACC records of 98.3%, and FAR records of 0.018, according to the acquired results, which were carried out utilising 48 features.

Figure 10 illustrates the convergence curve resulting from sample runs of the GOAMLP, E1GOAMLP, E2GOAMLP, E3GOAMLP, E4GOAMLP, E5GOAMLP, and E6GOAMLP algorithms against selected results from the SpamAssassin dataset. Figure 11 shows that the MOBGOA algorithm enhanced the classification accuracy by selecting a subset of 48 features. All the results in the matrices match those listed in Table 9.

Figure 11 presents the new proposed model MOBE2GOAMLP using confusion matrices due to the space constraints. It should be mentioned that their selection was

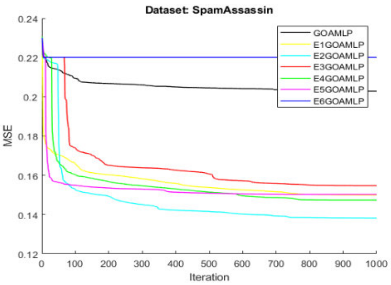


FIGURE 10. Convergence curve of proposed for training the spamassassin dataset.

TABLE 10. The classification results after using algorithms against selected subsets testing of the UK-2011.

Alg.	ACC	DR	FAR	MCC	PPV	NPV	SN	SP	F1	G
GOAMLP	92.2	93.4	0.088	0.84	0.90	0.94	0.93	0.91	0.92	92.3
E1GOAMLP	92.6	92.6	0.075	0.85	0.92	0.93	0.93	0.92	0.92	92.6
E2GOAMLP	**96.4**	**97.2**	**0.043**	**0.93**	**0.95**	**0.97**	**0.97**	**0.96**	**0.96**	**96.4**
E3GOAMLP	93.7	93.8	0.063	0.87	0.93	0.94	0.94	0.94	0.93	93.7
E4GOAMLP	94.5	94.3	0.053	0.89	0.94	0.95	0.94	0.95	0.94	94.5
E5GOAMLP	93.7	92.6	0.053	0.87	0.94	0.94	0.93	0.95	0.93	93.6
E6GOAMLP	92.5	93.4	0.083	0.85	0.91	0.94	0.93	0.92	0.92	92.5

TABLE 11. Comparison between the EGOAMLP and MOBEGOAMLP.

Dataset	EGOAMLP [41]			MOBEGOAMLP		
	ACC%	FAR	DR%	ACC%	FAR	DR%
SpamBase	96.9	0.037	97.2	97.5	0.033	98.1
SpamAssassin	98.1	0.012	97.8	98.3	0.018	98.3
UK-2011	95.6	0.053	96.6	96.4	0.043	97.2

arbitrary and that the intention was to showcase the most effective trainers using SpamAssassin.

C. UK-2011 WEBSPAM RESULTS

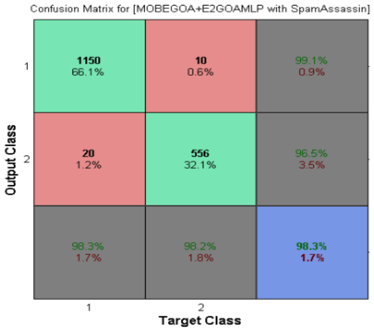
In this Scenario 3, the MOBGOA was first applied to the UK-2011 Webspam dataset to select suitable features from the dimensionality of the search space using the fitness function, resulting in 11 to 5 features as shown in Figure 2. The classification of the resulting features for training and the results obtained are presented in Figure 12. Classification results are displayed using the selected features extracted by MOBGOA training for each training set as presented in Table 10.

The proposed MOBE2GOAMLP algorithm is highlighted in bold text. As per Figure 12, the same results are described in a confusion matrix. From the discussion Subsections A and B, it is apparent that in the SpamAssassin and SpamBase datasets, using MOBGOA feature selection has improved the overall performance of the E2GOAMLP classifier. Figure 13 shows the convergence curve resulting from sample runs of the GOAMLP, E1GOAMLP, E2GOAMLP, E3GOAMLP, E4GOAMLP, E5GOAMLP, and E6GOAMLP algorithms against selected results of the UK-2011Webspam dataset. Figure 13 illustrations that the MOBGOA algorithm enhanced the classification accuracy by selecting a subset of 5 features. All the results in the matrices match those listed in Table 10.

TABLE 12. Comparison of the study's findings with previously published research.

Reference	Algorithm	FS.	DS.	ACC.
[35]	HHO KNN	57	SB	94.3%
[29]	BFA NBC	21	SB	95.14%
[25]	HWOAFPA KNN	48	SB	94%
[27]	WCSA SVM	26	SB	96.3%
[9]	GANB	38	SB	94.5%
[34]	FFOPSO KNN	10	SB	94.82%
[31]	SVM/NB	80	SA	97% , 98%
[30]	GA RWN	25	SA	92%
[48]	LSTM	-	SA	97.18%
[33]	HT/NB	-	SA	91.34% 91.64%
[32]	RF	-	UK	93%
[11]	CFS-PSO MLP/NB	10	UK	16.13% 8.23%
[49]	Bayesglm,Bagg, Boost, KKNN	-	UK	56.85% 71.44% 54.83% 83.6%
Our proposed	MOBEGOAMLP	15	SB	97.5%
		48	SA	98.3%
		5	UK	96.4%

Features Selected (FS), Dataset (DS), Not Available (-), SpamBase (SB), SpamAssassin (SA), UK-2011Webspam (UK), Accuracy (ACC)



	1	2	
1	1150 66.1%	10 0.6%	99.1% 0.9%
2	20 1.2%	556 32.1%	96.5% 3.5%
	98.3% 1.7%	98.2% 1.8%	98.3% 1.7%

FIGURE 11. Confusion matrices for mobeg2oamlp against the spamassassin dataset.

Figure 13 presents the new proposed model MOBE2GOAMLP using confusion matrices due to the space constraints. It should be mentioned that their selection was arbitrary and that the intention was to showcase the most effective trainers using the UK-2011 dataset.

D. THE ADVANTAGE OF THE MOBGOA

Table 11 compares the outcomes of analysing the resultant EGOAMLP models and the last MOBEGOAMLP models using three datasets. The evaluation comprised a comparison between the MOBEGOAMLP models, which used the chosen characteristics extracted by MOBGOA, and the EGOAMLP models, which used all features. ACC, DR, and FAR were used to gauge performance. The results clearly reveal that the most recent MOBEGOAMLP model exhibits a superior classification of ACC and DR across all data sets. These results offer the first proof that the EGOAMLP system is superior, with the last model showing a higher ACC and DR on all data sets, including SpamBase, SpamAssassin, and UK-2011Webspam.

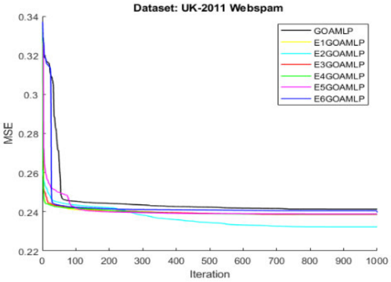


FIGURE 12. Convergence curve of proposed for training the UK-2011.

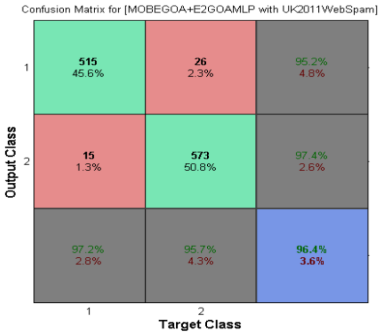


FIGURE 13. Confusion matrices for mobe2goamlp against the UK-2011 WEBSPAM dataset.

TABLE 13. Comparison between mobegoas and mobgoa at $A = 0.05$ on a two-tailed t-test.

Dataset	Model	MOB E1GOA	MOB E2GOA	MOB E3GOA	MOB E4GOA	MOB E5GOA	MOB E6GOA
SA	GOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
SB		0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
UK		1.16E-11	1.84E-109	2.15E-45	2.32E-44	1.19E-47	1.92E-31

E. COMPARISON OF THE RESULTS OF THIS STUDY AND THE PUBLISHED WORK

This section summarises the current state-of-the-art spam detection systems recorded in Table 12.

The overall results are much more satisfactory and fare well in comparison with the others, including datasets. This approach closely follows the performance of the best performing methods in the evaluation criteria. The records have been efficaciously classified through the proposed version as compared to the ones classified through other techniques.

F. EVALUATION USING T-TesT

In this section, we have analyzed the statistical analysis of the previous results in Table 13 and conducted the statistical t-test (T) to estimate the practical performance of the proposed algorithms compared with the standard algorithm (GOA). The proposed models' findings show statistically significant differences from those of the standard GOA method, with *P* values less than 0.05. In comparison to the standard algorithm GOA, the *P* values greater than 0.05 (underlined) are not significant. This table shows that, for all three datasets, the proposed models were always superior to the standard algorithm (GOA).

VII. CONCLUSION

This work introduces a novel method for SDS, the MOBGOA-trained EGOAMLP. It centres around the pertinence of a modern algorithm, referred to as MOBGOA, for preparing EGOAMLP. The MOB-EGOAMLP trained with the datasets had an accuracy of 97.5%, 98.3%, and 96.4% respectively. The results of this study show the highly positive impact of this approach on delivering a better SDS. Future research efforts will be to develop and extend an approach that can robustly be implemented in detecting other malicious attacks such as phishing and botnets.

REFERENCES

[1] D. M. Ablel-Rheem, "Hybrid feature selection and ensemble learning method for spam email classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1.4, pp. 217–223, Sep. 2020.

[2] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email classification research trends: Review and open issues," *IEEE Access*, vol. 5, pp. 9044–9064, 2017.

[3] A. Kumari, N. Agrawal, and U. Lilhore, "Clustering malicious spam in email systems using mass mailing," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 870–875.

[4] S. A. A. Ghaleb, M. Mohamad, S. A. Fadzli, and W. A. H. M. Ghanem, "E-mail spam classification using grasshopper optimization algorithm and neural networks," *Comput., Mater. Continua*, vol. 71, no. 3, pp. 4749–4766, 2022.

[5] S. A. A. Ghaleb, M. Mohamad, S. A. Fadzli, and W. A. H. M. Ghanem, "Spam classification based on supervised learning using grasshopper optimization algorithm and artificial neural network," *Commun. Comput. Inf. Sci.*, vol. 1347, pp. 420–434, Dec. 2021.

[6] M. Shuaib, S. M. Abdulhamid, O. S. Adebayo, O. Osho, I. Idris, J. K. Alhassan, and N. Rana, "Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification," *Social Netw. Appl. Sci.*, vol. 1, no. 5, p. 390, May 2019.

[7] S. A. A. Ghaleb, M. Mohamad, S. A. Fadzli, and W. A. H. M. Ghanem, "An integrated model to email spam classification using an enhanced grasshopper optimization algorithm to train a multilayer perceptron neural network," *Commun. Comput. Inf. Sci.*, vol. 1347, pp. 402–419, Dec. 2020.

[8] I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, "A combined negative selection algorithm-particle swarm optimization for an email spam detection system," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 33–44, Nov. 2015.

[9] O. M. E. Ebadati and F. Ahmadzadeh, "Classification spam email with elimination of unsuitable features with hybrid of GA-naive Bayes," *J. Inf. Knowl. Manage.*, vol. 18, no. 1, Mar. 2019, Art. no. 1950008.

[10] A. Karim, S. Azam, B. Shanmugam, and K. Kannoorpatti, "An unsupervised approach for content-based clustering of emails into spam and ham through multiangular feature formulation," *IEEE Access*, vol. 9, pp. 135186–135209, 2021.

[11] A. K. Singh and S. Singh, "Detection of spam using particle swarm optimisation in feature selection," *Pertanika J. Sci. Technol.*, vol. 26, no. 3, pp. 1–15, 2018.

[12] K. Wang, W. Mao, W. Feng, and H. Wang, "Research on spam filtering technology based on new mutual information feature selection algorithm," *J. Phys., Conf.*, vol. 1673, no. 1, Nov. 2020, Art. no. 012028.

[13] R. A. Atta, "Spam classification using genetic algorithm," *Iraqi J. Inf. Technol.*, vol. 9, no. 2, pp. 142–170, 2018.

[14] W. A. H. M. Ghanem and A. Jantan, "Novel multi-objective artificial bee colony optimization for wrapper based feature selection in intruction detectoin," *Int. J. Adv. Soft Comput. Appl.*, vol. 8, no. 1, pp. 70–81, 2016.

[15] B. Alatas and H. Bingol, "Comparative assessment of light-based intelligent search and optimization algorithms," *Light Eng.*, vol. 28, no. 6, pp. 51–59, 2020.

[16] H. Bingol and B. Alatas, "Chaotic league championship algorithms," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5123–5147, Dec. 2016.

[17] H. Bingol and B. Alatas, "Chaos based optics inspired optimization algorithms as global solution search approach," *Chaos, Solitons Fractals*, vol. 141, Dec. 2020, Art. no. 110434.

[18] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[19] S. A. A. Ghaleb, M. Mohamad, S. A. Fadzli, and W. A. H. M. Ghanem, *Integrating Mutation Operator Into Grasshopper Optimization Algorithm for Global Optimization*, vol. 25, no. 13. Berlin, Germany: Springer, 2021.

[20] A. Saad, S. A. Khan, and A. Mahmood, "A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design," *Swarm Evol. Comput.*, vol. 38, pp. 187–201, Feb. 2018.

[21] X. S. Yang, "Bat algorithm for multi-objective optimisation," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 5, pp. 267–274, 2012.

[22] E. G. Talbi, "A unified taxonomy of hybrid metaheuristics with mathematical programming, constraint programming and machine learning," *Stud. Comput. Intell.*, vol. 434, pp. 3–76, Dec. 2013.

[23] Z. Hassani, V. Hajihashemi, K. Borna, and I. S. Dehmajnoonie, "A classification method for E-mail spam using a hybrid approach for feature selection optimization," *J. Sci., Islamic Republic Iran*, vol. 31, no. 2, pp. 165–173, 2020.

[24] A. Jantan, W. A. H. M. Ghanem, and S. A. A. Ghaleb, "Using modified bat algorithm to train neural networks for spam detection," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 24, pp. 6788–6799, 2017.

[25] H. Mohmmadzadeh and F. S. Gharehchopogh, "A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study email spam detection," *Comput. Intell.*, vol. 37, no. 1, pp. 1–28, 2020.

[26] J. R. Méndez, T. R. Cotos-Yañez, and D. Ruano-Ordás, "A new semantic-based feature selection method for spam filtering," *Appl. Soft Comput.*, vol. 76, pp. 89–104, Mar. 2019.

[27] G. Al-Rawashdeh, R. Mamat, and N. H. B. A. Rahim, "Hybrid water cycle optimization algorithm with simulated annealing for spam E-mail detection," *IEEE Access*, vol. 7, pp. 143721–143734, 2019.

[28] T. Gangavarapu and C. D. J. B. Chanduka, *Applicability of Machine Learning in Spam and Phishing Email Filtering: Review and Approaches*. Amsterdam, The Netherlands: Springer, 2020.

[29] B. Ahmed, "Wrapper feature selection approach based on binary firefly algorithm for spam E-mail filtering," *J. Soft Comput. Data Mining*, vol. 2, no. 1, pp. 44–52, 2020.

[30] H. Faris, A.-Z. Ala'M, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, and H. Fujita, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks," *Inf. Fusion*, vol. 48, pp. 67–83, Aug. 2019.

[31] H. B. Ozkan and B. Can, "Analysis of adversarial attacks against traditional spam filters," in *Proc. Int. Conf. All Aspects Cyber Secur.*, 2019.

[32] J. Liu, Y. Su, S. Lv, and C. Huang, "Detecting web spam based on novel features from web page source code," *Secur. Commun. Netw.*, vol. 2020, pp. 1–14, Dec. 2020.

[33] S. Agrahari and A. K. Singh, "Disposition-based concept drift detection and adaptation in data stream," *Arabian J. Sci. Eng.*, vol. 47, no. 8, pp. 10605–10621, Aug. 2022.

[34] F. Soleimanian and S. K. Mousavi, "A new feature selection in email spam detection by particle swarm optimization and fruit fly optimization algorithms," *J. Comput. Knowl. Eng.*, vol. 2, no. 2, pp. 49–62, 2019.

[35] A. S. Mashaleh, N. F. B. Ibrahim, M. A. Al-Betar, H. M. J. Mustafa, and Q. M. Yaseen, "Detecting spam email with machine learning optimized with Harris hawks optimizer (HHO) algorithm," *Proc. Comput. Sci.*, vol. 201, pp. 659–664, Sep. 2022.

[36] E. Alba and J. F. Chicano, "Training neural networks with GA hybrid algorithms," in *Proc. Genetic Evol. Comput. Conf.* Berlin, Germany: Springer, 2004, pp. 852–863.

[37] S. Kang, J. Choi, and J. Choi, "A method of securing mass storage for SQL server by sharing network disks-on the Amazon EC2 windows environments," *J. Internet Comput. Services*, vol. 17, no. 2, pp. 1–9, Apr. 2016.

[38] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, Jan. 2016.

[39] N. Saidani, K. Adi, and M. S. Allili, "A semantic-based classification approach for an enhanced spam detection," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101716.

[40] W. Hijawi, H. Faris, J. Alqatawna, I. Aljarah, A. M. Al-Zoubi, and M. Habib, "EMFET: E-mail features extraction tool," 2017, *arXiv:1711.08521*.

[41] S. A. A. Ghaleb, M. Mohamad, S. A. Fadzli, and W. A. H. M. Ghanem, "Training neural networks by enhance grasshopper optimization algorithm for spam detection system," *IEEE Access*, vol. 9, pp. 116768–116813, 2021.

[42] W. A. H. M. Ghanem and A. Jantan, *Training a Neural Network for Cyber-attack Classification Applications Using Hybridization of an Artificial Bee Colony and Monarch Butterfly Optimization*, vol. 51, no. 1. Cham, Switzerland: Springer, 2020.

[43] W. A. H. M. Ghanem, S. A. A. Ghaleb, A. Jantan, A. B. Nasser, S. A. M. Saleh, A. Ngah, and A. C. Alhadi, "Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks," *IEEE Access*, vol. 10, pp. 76318–76339, 2022.

[44] Hopkins. (1999). *UCI Machine Learning Repository: Spambase Data Set*. Accessed: Nov. 1, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase

[45] SpamAssassin. (2005). *Spamassassin Public Corpus Kaggle*. Accessed: Nov. 1, 2021. [Online]. Available: https://www.kaggle.com/beatoa/spamassassin-public-corpus

[46] H. A. Wahsheh, M. N. Al-Kabi, and I. M. Alsmadi, "A link and content hybrid approach for Arabic web spam detection," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 1, pp. 30–43, Dec. 2012.

[47] H. A. Wahsheh, M. N. Al-Kabi, and I. M. Alsmadi, "A link and content hybrid approach for Arabic web spam detection," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 1, pp. 30–43, Dec. 2012.

[48] K. F. Rafat, Q. Xin, A. R. Javed, Z. Jalil, and R. Z. Ahmad, "Evading obscure communication from spam emails," *Math. Biosciences Eng.*, vol. 19, no. 2, pp. 1926–1943, 2021.

[49] A. Makkar and S. Goel, "Spammer classification using ensemble methods over content-based features," *Adv. Intell. Syst. Comput.*, vol. 547, pp. 1–9, Jun. 2017.

SANAA A. A. GHALEB received the bachelor's degree from the University of Aden, Yemen, in 2011, and the master's degree from Universiti Sains Malaysia, Malaysia, in 2017. She is currently pursuing the Ph.D. degree with the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin. Her research interests include technology-enhanced learning, instructional design and technology, computer networks and information security, cybersecurity, machine learning, artificial intelligence, swarm intelligence, and metaheuristic.

MUMTAZIMAH MOHAMAD was born in Terengganu, Malaysia. She received the bachelor's degree in information technology from Universiti Kebangsaan Malaysia, in 2000, the M.Sc. degree in computer science from Universiti Putra Malaysia, and the Ph.D. degree in computer science from Universiti Malaysia Terengganu, in 2014. She was a Junior Lecturer, in 2000. Currently, she is an Associate Professor with the Department of Computer Science, Faculty of Informatics and Computing (FIK), Universiti Sultan Zainal Abidin, Terengganu, Malaysia. She has published over 50 research articles in peer-reviewed journals, book chapters, and proceeding. She has appointed a reviewer and technical committee for many conferences and journals and worked as a researcher in several national funded Research and Development projects. Her research interests include pattern recognition, machine learning, artificial intelligence, and parallel processing.

WAHEED ALI H. M. GHANEM received the B.Sc. degree in computer sciences and engineering from Aden University, Yemen, in 2003, and the M.Sc. degree in computer science and the Ph.D. degree in network and communication protocols from Universiti Sains Malaysia, in 2013 and 2019, respectively. His research interests include computer and network security, cybersecurity, machine learning, artificial intelligence, swarm intelligence, optimization algorithm, and information technology.

ABDULLAH B. NASSER (Member, IEEE) received the B.Sc. degree from Hodeidah University, Yemen, in 2006, the M.Sc. degree from the Universiti Sains Malaysia, Malaysia, in 2014, and the Ph.D. degree from Universiti Malaysia Pahang, Malaysia, in 2018, all in computer science. He is currently an Assistant Professor with the Faculty of Computing, Universiti Malaysia Pahang. He has authored of many scientific papers published in renowned journals and conferences. His research interests include software testing and soft computing, specifically, the use of artificial intelligence methods (metaheuristic algorithms) for solving different software engineering problems.

MOHAMED GHETAS received the M.Sc. and Ph.D. degrees in computer science from Universiti Sains Malaysia. He is a Lecturer with the Faculty of Computer Science, Nahda University (USM). His research interests include cloud computing, fog-computing, robust optimization, evolutionary algorithm, federated learning, artificial neural networks, and deep learning.

AKIBU MAHMOUD ABDULLAHI received the B.A. degree in arabic language from Bayero University Kano, Nigeria, in 2011, the B.S. degree in information technology (IT) from Almadinah International University, Selangor, Malaysia, in 2016, the M.S. degree in instructional multimedia from University Sains Malaysia (USM), Penang, Malaysia, in 2017, and the Ph.D. degree in computer science from Taylor's University, Malaysia, 2021. From 2016 to 2018, he was an IT Help Desk Technician at Labtech International Ltd., Malaysia. He is currently a Lecturer with Albukhary International University, Kedah, Malaysia. His research interests include the data science, machine learning, learning analytics, and big data analytics.

SAMI ABDULLA MOHSEN SALEH received the B.Eng. degree in computer engineering from Hodeidah University, Yemen, in 2005, and the M.Sc. degree in electronic systems design engineering and the Ph.D. degree in computer vision and machine learning from Universiti Sains Malaysia, in 2013 and 2022, respectively. He was a Researcher at the Intelligent Biometric Group, School of Electrical and Electronic Engineering, Universiti Sains Malaysia. He is currently a Researcher with the Aerial Vehicle and Surveillance System Research Group, Aerospace Engineering School. His research interests include computer vision, deep learning, swarm intelligence, and soft biometrics. He has served as a Reviewer for several well-known conferences and international journals, such as *Pattern Recognition Letters* journal.

HUMAIRA ARSHAD received the master's degree in information technology from the National University of Science and Technology (NUST), Pakistan, and the Ph.D. degree from the School of Computer Science, Universiti Sains Malaysiais. She joined at the Faculty of Computer Sciences & IT, in 2004. She is an Associate Professor with the Department of Computer Sciences & IT, Islamia University of Bahawalpur, Pakistan. Her research interests include digital & social media forensics, information security, online social networks, cybersecurity, intrusion detection, reverse engineering, and semantic web.

ABIODUN ESTHER OMOLARA received the Ph.D. degree from the School of Computer Sciences, Universiti Sains Malaysia. Her research interests include computer and network security, cyber-security, cryptography, artificial intelligence, natural language processing, network and communication protocol, forensics, and the IoT security.

OLUDARE ISAAC ABIODUN received the Ph.D. degree in nuclear and radiation physics from the Nigerian Defence Academy, Kaduna, and the Ph.D. degree in computer science from the Universiti Sains Malaysia, Penang, Malaysia, with specialization in security and digital forensic. His research interests include artificial intelligence, robotics, cybersecurity, digital forensics, nuclear security, terrorism, national security, and the IoT's security.