Journal of Computer and Knowledge Engineering, Vol. 2, No.2, 2019.
DOI: 10.22067/cke.v2i2.81750

A New Feature Selection in Email Spam Detection by Particle Swarm Optimization and Fruit Fly Optimization Algorithms

Research Article

Farhad Soleimanian Gharehchopogh[1]* Seyyed Keyvan Mousavi[2]

Abstract. With the advent of the internet, along with email, and social networking, there are some new issues that have caused vulnerability of users against attackers. Internet users face a lot of undesirable emails and their data privacy and security is in danger. Spammers are often sent to users by intruders and sales markets, and most of the time they target spam, harassment, and abuse of user data. With increasing attacks on computer networks, attempts to rebuild computer networks and detect spam emails are important. Hackers use the identities of users by obtaining their personal information and account of users for malicious and subversive actions. Intruders are attempting to expose, remove, or change user information by opening encrypted information. Therefore, it is very important to detect spam in the early stages. In this paper, a new approach is proposed based on a hybridization of Particle Swarm Optimization (PSO) with Fruit Fly Optimization (FFO) to email spam detection. This paper shows a Feature Selection (FS) based on PSO, which decreases dimensionality and improves the accuracy of email spam classification. The PSO searches the feature space for the best feature subsets. Experiments results on the public spambase dataset show that the accuracy of the proposed model is 92.21%, which is better in comparison with others models, such as PSO, Genetic Algorithm (GA), and Ant Colony Optimization (ACO).

Keywords. Email spam detection, Feature selection, Particle swarm optimization, Fruit fly optimization.

1. Introduction

Email is one of the easiest ways of communicating to the online environment. One of the main popularities of the email is because text and images can be both sent. Unfortunately, despite the great benefits of the internet environment, some of intruders, online stores, social networks, and news services are sending spam email to users, and the user's mailbox is filled with a lot of spam that is very frustrating for users [1]. There are several ways to reduce spam, one of which is the use of anti-spam [2]; this means that software and tools to prevent spam from being used must be used. The two most important methods in which users can detect spam are the knowledge engineering and machine learning algorithms. Knowledge engineering means that internet and network protocols are used to email spam detection, and machine learning algorithms use training and testing for detection which is successfully used for email spam detection [3].

Spam is known as an unwanted email that contains viruses and spyware sent for fraudulent and malicious purposes along with advertising purposes. Signs such as specific keywords, numbers, and symbols help spam detection. Most spammers use certain phrases when sending email and use unique words in the email body [4]. Meanwhile, e-mail companies can prevent users from installing and using email spam detection programs to generate and send them to users. In most cases, opening spam emails leads to disrupt and slow down the system.

Spams can steal user's information such as their username and password by social engineering techniques, fake links, and fake sites. Identifying and blocking spam is one of the key issues in cyber security, which can greatly reduce the effect of this undesirable internet phenomenon and the security challenge of email service. Identifying the hidden patterns of spam by data mining and machine learning methods makes the emails received accurately categorized into two categories of spam and non-spam.

In order to deal with the problem of email spam, many different models have been proposed. In [5], a hybrid model of PSO and Negative Selection Algorithm (NSA) for email spam detection is proposed. In this model, the spambase dataset has been used in the training and testing phases to optimize the PSO for data training and to use the NSA to test the data. The results showed that the accuracy of the hybridization model is 91.22% which is better than the PSO, NSA, Naïve Bayes (NB), and Support Vector Machine (SVM). The NB and SVM are 79.3% and 90.00%, respectively.

A hybrid model of Differential Evolution (DE) and NSA is proposed for detecting spam [6]. In NSA-DE model, the spambase dataset has been used in the training and testing phases. In NSA-DE, DE for training data and NSA to test data is used. The obtained results showed that the precision of the hybridization model is equal to 65.14%, which is more accurately compared to NSA and DE.

A new e-mail detection approach based on an improved NSA called combined clustered NSA and fruit fly optimization (CNSA–FFO) has been proposed [7]. In the hybrid model, the NSA has been improved based on FFO. In this model, the hybridization of NSA with k-means and FFO was used to improve NSA. The results showed that the CNSA-FFO is more accurate than the NSA and the NSA-PSO. The percentage of accuracy the CNSA-FFO model is 93.88%.

Manuscript received July, 6, 2019; accepted February, 11,2020.

[1]* F.Soleimanian Gharehchopogh; Assistant Professor, Urmia, 2km Airport, Islamic Azad University, Urmia, Faculty of Engineering, Department of Computer Engineering.
Email: bonab.farhad@gmail.com.
[2] K. Mousavi, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, IRAN.

In [8] Header Based Email Spam Detection Framework using Support Vector Machine (SVM) has been proposed. Explements have done on two email datasets (Anomaly Detection Challenges and Cyber Security Data Mining from website). There are five phases in the model which are data collection, data pre-processing, features selection, classification and detection. The SVM has proven to be a successful classifier which produced above 80% accuracy rate for both datasets.

In [9], SVM is proposed for spam detection in order to appropriately search for the optimal parameters. Experimental results showed that the proposed model outperformed all the others proposed models on the spambase dataset employed. Accuracy of 95.87 and 94.06% was obtained for training and testing sets, respectively. The 94.06% testing accuracy showed an improvement of 3.11% over the best reported model.

In the present work, a hybrid model based on FFO [10] and PSO [11], which are metaheuristic algorithms is proposed for email spam detection. A hybridization of FFO is used to optimize the optimal vectors and to solve the problems of the PSO. The combination of PSO with FFO to maximize the coverage of the space search to solve problem in email spam detection. The main advantage of FFO is that it has the ability to optimize the solution with global search solution space. The features must be selected by the particles in the environment, and the Feature Selection (FS) must be in the neighborhood of each other and the accuracy of classification is high. The overall process of the proposed model consists of three steps: the particle distribution stage, the stage of FS, and the stage of data classification.

The high number of features not only does not necessarily lead to high accuracy, but in some cases leads to a loss of accuracy, so reducing the feature can increase accuracy. Reducing the features can lead to increased classification accuracy by eliminating unnecessary features [12]. The FS is one of the most important steps that increases the efficiency of classifying samples [13].

In the following, the overall structure of this paper is as follows. In the Section 2, the basic algorithms are explained. In the Section 3; the model is proposed. In the Section 4, the proposed model is evaluated and compared with other models, and finally, in the Section 5, conclusions are drawn and the future work is presented.

2. Basic Algorithms

In this section, two algorithms of FFO and PSO are describe.

2.1. Fruit Fly Optimization Algorithm

FFO is defined based on fruit fly eating behavior. The fruit fly is stronger than other insects, and has a stronger sense of smell and vision, so it can detect the smell of fruit in the air. This insect, after smell of fruit and after approaching the position of the fruit, can find the exact position of the fruit using its sense of sight and working with others. Figure (1) shows the structure of food search by the fruit fly [10].

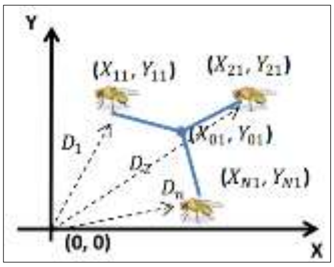

Fig. 1: Food search by the fruit fly [8]

This algorithm consists of several positions. The steps of the FFO are as follows [10]:

1) The fruit fly position is randomly initialized.
2) Determine the direction and distance of the search for flies randomly according to Eq. (3) and Eq. (4).

$$X_{axis} = lower_{bound} + (upper_{bound} - lower_{bound}) * rand() \tag{1}$$

$$Y_{axis} = lower_{bound} + (upper_{bound} - lower_{bound}) * rand() \tag{2}$$

$$X_i = X_{axis} + Random\ Value \tag{3}$$

$$Y_i = Y_{axis} + Random\ Value \tag{4}$$

3) Since the position of the fruit is not known, then the distance to the source must first be calculated, and then the odor intensity (S) is calculated. This is the distance from the inverse in which the more the smell is, the distance is less:

$$D_i = \sqrt{X_i^2 + Y_i^2} \tag{5}$$

$$S_i = 1/D_i \tag{6}$$

4) The amount of odor intensity is replaced by the fitness function. Then the smell of the existing position is calculated according to Eq. (7).

$$Smell_i = Function(S_i) \tag{7}$$

5) The fruit fly with the highest intensity of smell (find the highest value) is found from the congestion according to Eq. (8).

$$[Best\ Smell\ Best\ Index] = Max(Smell) \tag{8}$$

6) If the odor intensity in each replicate is better than the current value, then the best amount of odor intensity and coordinates *X* and *Y* are stored. At this time, the fruit fly can move towards the fruit with respect to its power of sight:

$$\begin{aligned} Smell\ Best &= Best\ Smell \\ X_{axis} &= X(best\ index) \\ Y_{axis} &= Y(best\ index) \end{aligned} \tag{9}$$

7) Steps 2 to 6 of the optimization are repeated until the condition is met.

2.2. Particle Swarm Optimization

The PSO is a population-based algorithm in which particles form a swarm (population) [11]. The population moves in the space of the problem and, based on their individual experiences and collective experiences, are trying to find the optimal solution to the search space. The PSO, as an optimization algorithm, provides a population-based search in which each particle changes its position over time. In PSO, particles move in a multi-dimensional search space from possible problem solving. In this space, an evaluation criterion is defined and a quality assessment of the problem solution is made. The change of the mode of any particle in a group is influenced by its own experiences or the knowledge of its neighbors, and the search behavior of a particle in the group is influenced by other particles. This simple behavior makes it possible to find optimal areas of search space. Therefore, in PSO, each particle, as soon as its optimal position is found, correctly informs other particles, and each particle decides on the basis of the values obtained for the cost function with a certain probability to follow other particles. The search in the problem space is based on previous particle knowledge. This action does not make all the particles too close to each other and can effectively solve continuous optimization problems.

In the PSO, group members are randomly created in the problem space, and the search begins to find the optimal answer. In the general structure of the search, each particle follows the particle that has the optimal fitness function, while also not forgetting its own experience and following the condition in which it has the best fitness function. Therefore, in each algorithm, each person changes his next position according to two values: First, the best position that a particle has had (pbest), and the best situation ever created by the entire population, and in fact the best pbest is in the all of population (gbest). Conceptually, the pbest for each individual is actually the biological memory of that person. That gbest is the same as the general knowledge of the population, and when people change their position based on gbest, they are actually trying to bring their knowledge to the knowledge level of the population. Conceptually, the best particle of a group is related to each particle of the group. The next position for each particle is determined by Eq. (10) and Eq. (11) [11].

$$v_{id}^{t+1} = w.v_{id} + c_1.r_1.(P_{best_i} - x_i) + c_2.r_2.(g_{best_i} - x_{id}) \tag{10}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{11}$$

In Eq. (10), c_1 and c_2 are learning parameters. Where x_{id} is the binary bits, $i = 1,2,\dots,n$ (n is set to be the total number of particles), $d = 1,2,\dots,m$ (m is the dimensionality of the data). Parameters r_1 and r_2 are a function for generating random numbers in the range [0,1]. x_{id} is the current position and v_i is the speed of movement of individuals w is a control parameter that controls the effect of the current velocity (v_{id}) on the next speed and creates a balance between the ability of the algorithm to search locally and search globally and, thus, on average, we will respond in less time. Therefore, for the optimal performance of the algorithm in search space, the parameter w is defined by Eq. (12) [11].

$$w = w_{Max} - \frac{((w_{Max} - w_{Min}) \times i)}{i_{Max}} \tag{12}$$

In Eq. (12) i_{max} represents the maximum number of repetitions of the algorithm and the parameter i of the repeater counter to find the optimal answer. In Eq. (12), parameters w_{max} and w_{min} are the initial value and the final value of the inertial mass, respectively. During the execution of the algorithm. These inertia weights vary linearly from 0.9 to 0.4 during the program execution. If w to be equal to large values, it leads to global searches and if w to be equal to small values, it leads to local searches. In order to balance the local and global search, it is necessary to reduce the inertial weight evenly during the execution of the algorithm. Therefore, by lowering the value of w, more searches occur locally and around the optimal answer.

3. Proposed model

The proposed model is a hybridization of PSO and FFO. In the proposed model, FFO is used to optimize the PSO. It should be noted that the PSO is weak and can capture the search in the trap of local optimizations. Therefore, this paper proposes the use of FFO to improve the performance of PSO and to reduce its weaknesses. It has also been proven that FFO works well in avoiding traps in local optimizations. FFO has the ability to escape local optimizations and, in most cases, converges to the optimal point. If the answer lies in the optimal locale, the optimal value for the revelation function is not found.

The first part in Eq. (10) represents the coefficient of the current velocity of the particle. The second part represents the movement of the particle towards the best of personal knowledge, and the third part is the particle movement towards the best group knowledge, and the search space is gradually shrinking and the best part is formed around the best of the particles to get the best answer. But, for particles in which the second and third parts of Eq. (10) are 0, the particle moves in the direction of its previous motion vector, and the rest of the particles converge to this particle, and so the algorithm converges quickly to a local optimal. FFO is used to solve the problem of PSO.

In the proposed model, the Eq. (13) is used to binary the PSO. The particle position is calculated after the update by Eq. (13). If the v_{id} value is greater than the random value (rand). In this case, the position value is equal to 1 (FS). In contrast, if the value of v_{id} is smaller than the random value (rand), the position value will be 0 (not FS). In each dimension, a particle value {1} indicates the FS can contribute for the next iteration. On the other hand, a particle value of {0} is not required as a pertinent for next iteration. Figure (2) demonstrates vector of particle for feature selection

$$\begin{aligned} & s(v_{id}(t+1)) = \frac{1}{1+e^{-v_{id}}} \\ & if\, rand < s(v_{id}(t+1))\, then\, x_{id}(t+1) = 1 \\ & else\, x_{id}(t+1) = 0 \end{aligned} \tag{13}$$

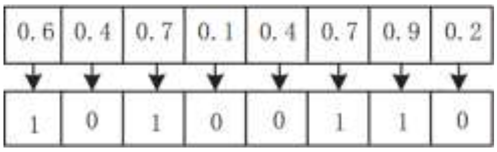

Fig. 2. Vector of particle for feature selection

In the FFO, the initial position of the flies is determined based on the values of the dataset. Improve the particle position update using FFO. In the PSO, some particles can be trapped in the optimal localization and cannot be removed in several repetitions from a non-finite point, and so all particles move to a non-finite point. Therefore, Eq. (14) is used to update the particle position. In the early iterations, a larger search scope is recorded to warrant that the fruit flies are able of searching food sources in a wide area and the global exploration power is elevated.

$$x_i = X_{axis} + w.v_{id}.p_{best(i)} \tag{14}$$

$$v_i = Y_{axis} + w.v_{id}.g_{best(i)} \tag{15}$$

To improve the position of the particles, the current position of the flies, as well as the control parameter (w) and vector v are used. X_{axis} and Y_{axis} are coordinates of flies in FFO. The purpose of w and v is to use particle positioning in the entire space to detect optimal positions. Also, the parameters of $pbest$ and $gbest$ are used to distribute the knowledge and the general knowledge. With $pbest$ and $gbest$ agents, poor particles also participate in the upgrade, so the chance to find optimal points with the hybridization of unplanned points is higher.

In Eq. (14) and Eq. (15), the parameter w has a significant effect on the convergence behaviour of the algorithm. If value of w is high then ability of the algorithm to find the global point in the search increases and the ability to locate the local point will be weak. The effect of the previous speed on the current speed of the algorithm can be controlled by setting the w parameter. The value w of the w_{max} value will be at least w_{min} in a linear repetitive process. The ability to search in a repeat algorithm process is strong, and this algorithm will be able to search in a large space of the answer, and new areas will be constantly reviewed to find the answer. From the perspective of repetitive repetition, the algorithm gradually reduces the scope of its search to a region, thereby, increasing the rate of convergence.

In the PSO, each particle has two positions and velocity vectors are updated in each repetition. The position vector of each particle contains the optimal value of the problem. In this paper, the components of the position vector of each particle are the same values of the dataset. In the first step, the FFO was used to determine the vectors of the velocity and position of each particle.

3.1. Feature Selection

FS is one of the approaches to improving accuracy and speed in machine learning algorithms. In the past few years, numerous studies have been carried out on email spam detection in the field of FS. The research results in the field of reduced features have shown that choosing a set below the initial characteristics can increase the accuracy of machine learning algorithms. These algorithms try to reduce the dimensions of the data by selecting a subset of the initial properties [14, 15]. In these algorithms, it is searched to find the subset with the minimum possible size of the features appropriate for the application. In most cases, data analyses such as classification on a reduced space are better than the original space.

Email spam including a set of numeric or categorical features $(f(1), f(2), f(3), \dots, f(n), f(n+1))$ where n shows predictive features and h(n+1) is a class of emails, namely spam and non-spam. FS is based on the PSO. In the proposed model, Eq. (15) is used to find the best position for $pbest$. In Eq. (16), x is the position of the particle k^{th}, also the max and min are highest and smallest form a particle. N is number of particles. In the mixed-purpose model of Eq. (16), finding the optimal points in the search space. The optimal spots in space are the same features that are selected for the classification stage. The features used in the preceding paper include numerical values. Features are chosen to reduce the value of d between them.

$$D_{ij} = \left[\sum_{k=1}^{N} \left(\frac{x_k^i - x_k^j}{\max_k - \min_k} \right)^2 \right]^{1/2} \tag{16}$$

The steps in the proposed model are as follows:

Table 1: Proposed Model Process

1) Initial population creation and distribution in space using FFO.
2) Calculate the initial position of the flies using Eq. (5)
3) Updating the particle position using Eq. (14) and Eq. (15)
4) Calculate the position of each particle
5) If the particles move towards the optimum global point, the best index will be saved.
6) If the particles do not move to the optimum global point, the velocity of the particles changes based on the weight of the inertia.
7) FS based on the proximity of particles using Eq. (16)
8) Training step
9) Build a training model and check the amount of features
10) Build classes and recognize features
11) Classification of samples
12) Data testing
13) Evaluation of new samples
14) Maximum program repetition
15) End

In Figure 3, the proposed model flowchart is shown. Flowchart as the proposed model method consists of initial population generation, updating, FS, sample training, sample testing, and classification. In the proposed model, initial population is produced by FFO. Primary population includes the amount of spambase dataset properties. Proposed model consists of 30 particles and each particle of 57 binary bits.

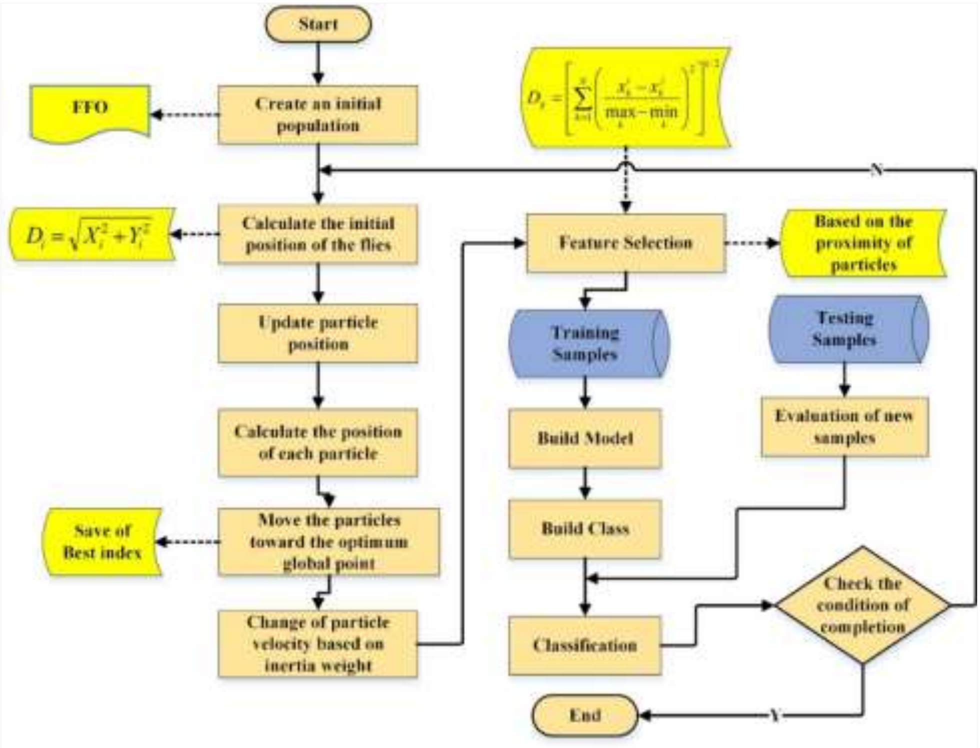

Fig. 3. Flowchart of the proposed model

3.2. Data Classification

In the proposed model, the distance criterion is used according to the FFO to classify the samples. In this respect, the features are numeric, the best criterion to use is to distance. Assuming that is used two particle position vectors $X_b=(x_{b1},x_{b2},x_{b3},...,x_{bn})$ (a vector with different properties) and $Y_b=(y_{b1},y_{b2},y_{b3},...,y_{bn})$ (a vector with different characteristics), and also the position of the best smell in FFO $X_w=(x_{w1},x_{w2},x_{w3},...,x_{wn})$ and $Y_w=(y_{w1},y_{w2},y_{w3},...,y_{wn})$. If the distance criterion is defined by Eq. (17).

$$d(x,y)=\sqrt{\sum_{j=1}^{n}((x_{bj})^2-(x_{wj})^2)+\sum_{j=1}^{n}((y_{bj})^2)-((y_{wj})^2)} \tag{17}$$

The distance between the features for each vector is calculated, and then the vectors that are more similar are placed in a class.

4. Evaluation and Results

In this section, the proposed model tests are performed on a system with Intel (R) Core (TM) i7-4510U @ 2.00 GHz CPU and 6 GB memory. In this paper, the most important criteria chosen for prediction is accuracy as it is the most important criterion in detection. In this paper, the accuracy of classifier acts as a significant task in FS, because the accuracy of email spam detection is based on classification accuracy that reduces the rate of errors, so the parameters of PSO such as r_1 and r_2 are between 0 and 1. The population size = 30, also, $C_1$and C_2 are set to 2 and the weight values are 1.

The evaluation of the proposed model is done by the division of the dataset using a stratified sample method with 80% training set and 20% testing set to check the efficiency of the new model on an unseen data. The training set is applied in the construction of the model by training the dataset on both models while evaluating the capability of the model with the testing set.

***Precision*:** Precision is defined as the ratio of correctly assigned category C samples to the total number of samples classified as category C as in Eq. (18). *Recall*: The ratio of the number of positive samples correctly detected to all positive samples, that is, Eq. (19). *F1*: A hybridization of precision and recall criteria that can be calculated according to Eq. (20). This criterion is, in fact, the harmonic average of the accuracy and recall parameters, namely, Eq. (20). *Accuracy*: The ratio of correct samples to all samples hit by the model; that is, Eq. (21).

$$\text{Precision(P)}=\frac{\text{TP}}{\text{TP+FP}} \tag{18}$$

$$\text{Recall(R)}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{19}$$

$$\text{F1}=\frac{2\times\text{P}\times\text{R}}{(\text{P}+\text{R})} \tag{20}$$

$$\text{Accuracy}=\frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \tag{21}$$

$$\text{ErrorRate}=1-\text{Accuracy} \tag{22}$$

The TP parameter (True Positive) represents the number of samples that are positive and accurately predicted. The FP (False Positive) parameter represents the number of false positive samples expected to be positive. The FN (False Negative) parameter represents the number of instances that are false as negative categories. The TN parameter (True Negative) represents the number of samples that are negative and well-predicted.

Table 2: Evaluation of the proposed model based on FS and 100 iterations

#FS	Models	Criteria					
		Precision	Recall	F-Measure	Accuracy	Error Rate	Time (Sec)
10	Proposed Model	93.68	95.07	94.51	94.82	5.18	71
	FFO	89.25	89.74	89.49	89.15	10.85	38
	PSO	85.67	86.03	85.83	85.23	14.77	42
12	Proposed Model	93.15	94.67	93.92	94.05	5.95	76
	FFO	88.92	89.10	89.01	89.05	10.95	40
	PSO	85.32	85.65	85.48	85.09	14.91	43
18	Proposed Model	92.90	93.08	92.98	93.66	6.34	79
	FFO	88.65	88.74	88.69	88.69	11.31	42
	PSO	85.19	85.28	85.23	84.78	15.22	45
22	Proposed Model	92.45	93.73	93.08	93.51	6.49	80
	FFO	88.45	88.51	88.48	88.31	11.69	44
	PSO	84.67	84.82	84.74	84.39	15.61	48
25	Proposed Model	91.70	92.10	91.89	92.94	7.06	83
	FFO	88.13	88.92	88.52	88.06	11.94	46
	PSO	84.28	84.76	84.52	84.15	15.85	50
30	Proposed Model	91.32	91.99	91.65	92.36	7.64	85
	FFO	88.02	88.68	88.35	87.93	12.07	49
	PSO	83.89	84.16	84.02	84.02	15.98	54
32	Proposed Model	91.08	92.01	91.54	92.04	7.96	89
	FFO	87.84	88.12	87.98	87.62	12.38	51
	PSO	83.59	83.96	83.77	83.94	16.06	57
36	Proposed Model	90.83	91.11	90.96	91.86	8.14	92
	FFO	87.58	87.79	87.68	87.43	12.57	53
	PSO	83.39	83.92	83.65	83.65	16.35	59
40	Proposed Model	90.51	91.33	91.93	91.22	8.78	95
	FFO	87.37	87.74	87.55	87.19	12.81	57
	PSO	83.26	83.79	83.52	83.44	16.56	62
42	Proposed Model	90.02	91.65	90.84	90.79	9.21	97
	FFO	87.15	84.03	85.56	87.05	12.95	60
	PSO	83.08	83.61	83.34	83.26	16.74	65
45	Proposed Model	89.82	91.49	91.64	90.03	9.97	100
	FFO	86.89	87.16	87.02	86.91	13.09	63
	PSO	82.91	83.25	83.08	83.14	16.86	69
48	Proposed Model	89.26	90.36	89.75	90.73	9.27	102
	FFO	86.56	86.93	86.74	86.69	13.31	65
	PSO	82.66	82.94	82.80	82.96	17.04	71
50	Proposed Model	88.53	89.02	88.72	89.91	10.09	105
	FFO	86.32	86.81	86.56	86.35	13.65	67
	PSO	82.49	82.79	82.64	82.79	17.21	72
52	Proposed Model	87.61	88.31	87.95	89.13	10.09	109
	FFO	86.18	86.74	86.46	86.23	13.77	69
	PSO	82.35	82.91	82.63	82.58	17.42	74
54	Proposed Model	87.17	88.50	87.83	88.17	11.83	113
	FFO	85.89	86.25	86.07	86.12	13.88	71
	PSO	82.09	82.68	82.38	82.36	17.64	76
55	Proposed Model	86.91	87.03	86.97	87.30	12.70	118
	FFO	85.59	86.16	85.87	85.93	14.04	72
	PSO	81.90	82.03	81.96	81.97	18.03	76
57	Proposed Model	86.49	88.15	87.31	87.02	12.98	123
	FFO	85.40	85.63	85.51	85.72	14.28	74
	PSO	81.52	81.69	81.60	81.57	18.43	78

Table 3: Evaluation of proposed model based on FS and 200 iterations

#FS	Models	Criteria					
		Precision	Recall	F-Measure	Accuracy	Error Rate	Time (Sec)
10	Proposed Model	95.23	95.89	94.53	97.15	2.85	86
	FFO	91.65	92.79	92.22	91.48	8.52	41
	PSO	88.92	89.03	88.97	88.82	11.18	43
12	Proposed Model	95.07	96.15	95.62	96.83	3.17	89
	FFO	91.38	91.69	91.53	91.18	8.82	42
	PSO	88.74	89.05	88.89	88.61	11.39	44
18	Proposed Model	94.73	95.08	94.90	96.24	3.76	91
	FFO	90.82	91.07	90.94	91.05	8.95	45
	PSO	88.32	88.73	88.52	88.35	11.65	46
22	Proposed Model	94.68	95.26	94.96	95.92	4.08	95
	FFO	90.42	90.66	90.54	90.51	9.49	48
	PSO	88.12	88.53	88.32	88.49	11.51	50
25	Proposed Model	93.62	94.51	94.06	95.71	4.29	102
	FFO	89.90	90.07	89.98	90.26	9.74	51
	PSO	87.96	88.15	88.05	88.14	11.86	54
30	Proposed Model	93.21	93.26	93.23	95.49	4.51	105
	FFO	89.62	89.93	89.77	89.92	10.08	54
	PSO	87.76	87.83	87.79	87.86	12.14	59
32	Proposed Model	92.49	93.84	93.05	94.86	5.14	112
	FFO	89.44	89.62	89.53	89.77	10.23	56
	PSO	87.31	87.81	87.56	87.39	12.61	61
36	Proposed Model	92.31	94.08	93.18	94.06	5.94	118
	FFO	89.19	89.82	89.50	89.26	10.74	60
	PSO	87.11	87.35	87.23	87.19	12.81	65
40	Proposed Model	91.90	92.37	92.35	93.79	6.21	122
	FFO	88.82	89.13	88.97	88.65	11.35	63
	PSO	86.53	86.81	86.67	86.41	13.59	68
42	Proposed Model	91.67	92.62	92.14	93.56	6.44	128
	FFO	88.53	88.72	88.62	88.37	11.63	65
	PSO	86.39	86.91	86.65	86.23	13.77	70
45	Proposed Model	91.34	92.08	91.70	93.44	6.56	130
	FFO	88.28	88.57	88.42	88.24	1.76	67
	PSO	85.75	85.90	85.82	85.93	14.07	73
48	Proposed Model	90.91	93.47	92.17	93.17	6.83	132
	FFO	88.13	88.71	88.42	88.11	11.89	70
	PSO	85.44	85.62	85.53	85.66	14.34	75
50	Proposed Model	90.48	91.16	90.81	93.05	6.95	138
	FFO	87.83	87.98	87.90	87.91	12.09	72
	PSO	85.26	85.47	85.36	85.42	14.58	77
52	Proposed Model	90.12	91.54	90.82	92.98	7.02	143
	FFO	87.76	87.91	87.83	87.62	12.38	74
	PSO	85.14	85.38	85.26	85.17	14.83	79
54	Proposed Model	89.02	90.21	89.61	92.63	7.37	148
	FFO	87.35	87.70	87.52	87.26	12.74	75
	PSO	84.79	84.92	84.85	84.62	15.38	80
55	Proposed Model	89.54	90.62	90.07	92.42	7.58	150
	FFO	87.18	87.71	87.44	87.19	12.81	76
	PSO	84.26	84.69	84.47	84.33	15.67	81
57	Proposed Model	89.16	90.37	89.76	92.21	7.79	156
	FFO	86.82	86.91	86.85	86.80	13.20	79
	PSO	83.73	83.85	83.79	83.15	16.85	83

4.1. Dataset

The spambase dataset is a collection of emails that contain 4601 samples and 58 features, compiled by Hopkins and colleagues [16]. The spambase dataset contains two spam classes with 1813 samples (39.4%) and non-spam with 2788 samples (60.6%). The first 48 features of the spambase dataset are taken from the repetition of certain particular words. The next six features are the percentage of the occurrence of a special character, such as “;”, “(“, “[“, “$”, “#”. The next three features represent the different metrics of repeating letters in the message text. Finally, the last class label property which indicates whether a spam sample was or that non-spam sample.

4.2. Evaluation based on Iterations

In Table (2), the results of the evaluation of the proposed model based on the FS and with 100 iterations have been shown that the FS is very effective in increasing the accuracy of detection. Proposed model based FS is run 10 times and the average of 10 runs is calculated as percentage of accuracy. If the number of features is lower, the percentage accuracy of the proposed model increases as by reducing features, finding the same properties in less time and better detection accuracy is better. For example, the percentage of accuracy with 10 features in proposed model is 94.82% and with 57 features it is 87.02%. Also, the percentage of accuracy with 10 features in FFO and PSO is 89.15% and 85.23% respectively. In general, if the number of features is 57, the percentage of the proposed model is 87.02%. It could be noticed that the proposed model achieved an improvement of 5.45% in comparison with PSO. The error rate of the proposed model (12.98%) is better than the FFO (14.28%) and PSO (18.43%) for 57 features.

In Table (3), the results of the evaluation of the proposed model are shown based on FS and with 200 iterations. Simulation results were also evaluated with higher iterations, but 200 iterations were best. Table (3) shows that increasing the iteration of the hybrid model is very effective in increasing the accuracy of detection. The results show that in case of 200 iterations, if the number of features is less, the percentage of the proposed model's accuracy increases. In general, if the number of features is 57, the percentage of the proposed model is 92.21%. This percentage is derived from the total number of features, therefore, this percentage is considered as the main percentage for the classification of the proposed model. The percentage of accuracy with 10 features in proposed model is 97.15%. Also, the percentage of accuracy with 10 features in FFO and PSO is 91.48% and 88.82%, respectively. It could be noticed that the proposed model achieved an improvement of 9.06% in comparison to PSO for 57 features.

The proposed model gradually reduces the scope of your search to a range, thus, increasing the convergence rate. The proposed model converges in 200 iterations. In the proposed model, when the current optimal answer does not show any improvement in continuous iterations, it assumes that the necessary convergence is achieved and the execution of the program ends.

In Figure (4), the comparison diagram of the proposed model, FFO and PSO is shown based on 100 iterations. In Figure (5), the comparison diagram of the proposed model, FFO and PSO is shown based on 200 iterations. In Figure (6), the comparison diagram of the proposed model is shown based on the various iterations. Figure (6) shows that the accuracy of the proposed model is more in 200 iterations. There is a significant increase in accuracy from 87.02% to 92.21% with 200 iterations and 57 features in proposed model.

Figure (7) shows 10 runs for 200 iterations of the proposed model. The results obtained from Figure (7) show that the proposed model has different results with each run, and the best percentage of accuracy in the proposed model for 200 iterations is 92.21%, which occurred in the fifth mode. It is worth noting that in 10 executions, the proposed model in most cases has a high result of 92%, with the highest percentage being considered as the final output. This accuracy may have occurred due to good convergence. The higher the number of features, the greater the number of local optimization points of the search space; therefore, Figure (7) clearly shows that the proposed model was able to obtain the best percentage of accuracy from the search space.

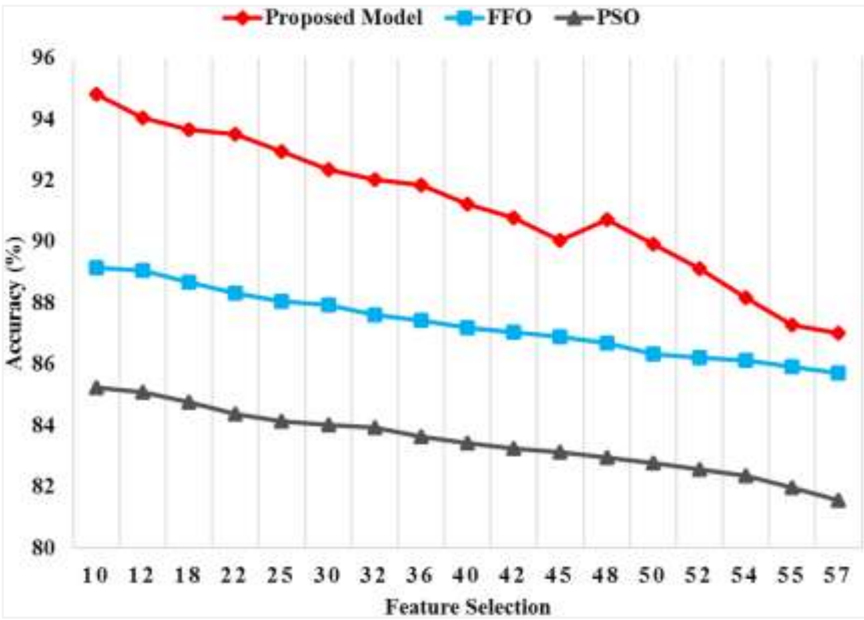

Fig. 4: comparison diagram of the proposed model, FFO and PSO based on 100 Iterations

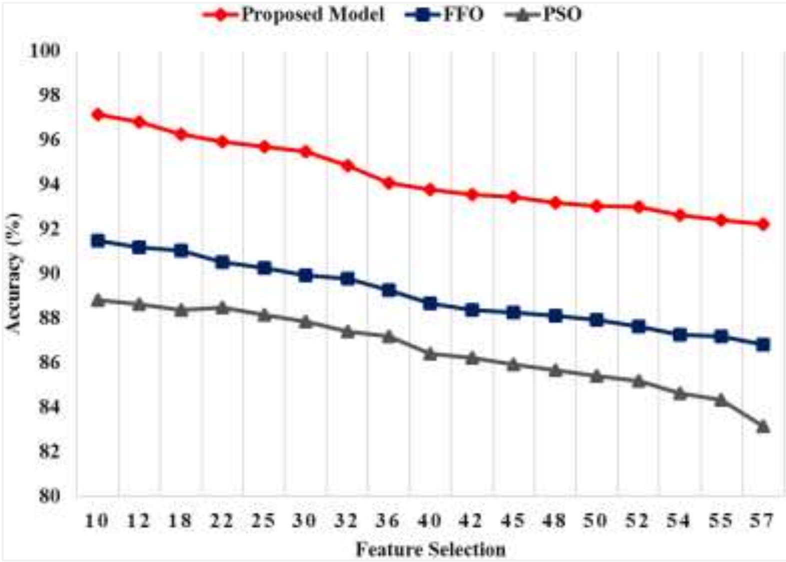

Fig. 5: comparison diagram of the proposed model, FFO and PSO based on 200 Iterations

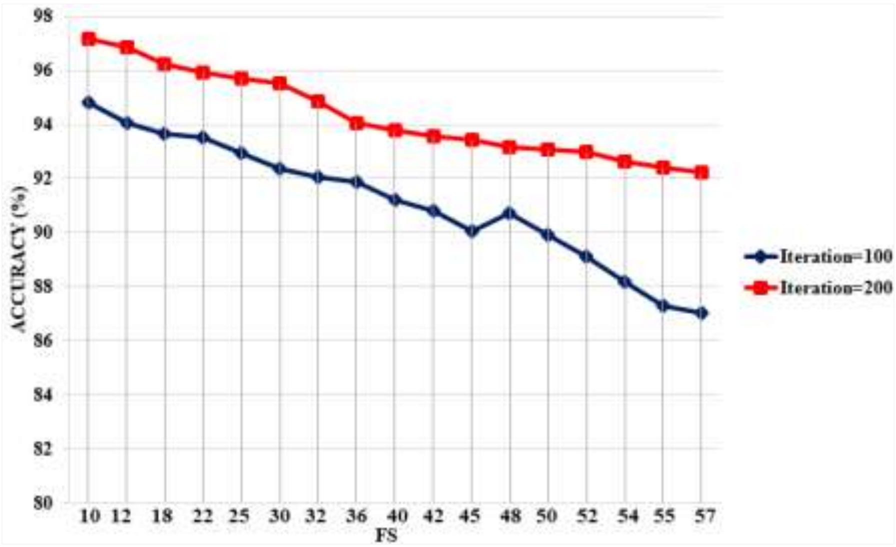

Fig. 6: Comparison diagram of the proposed model based on different iterations

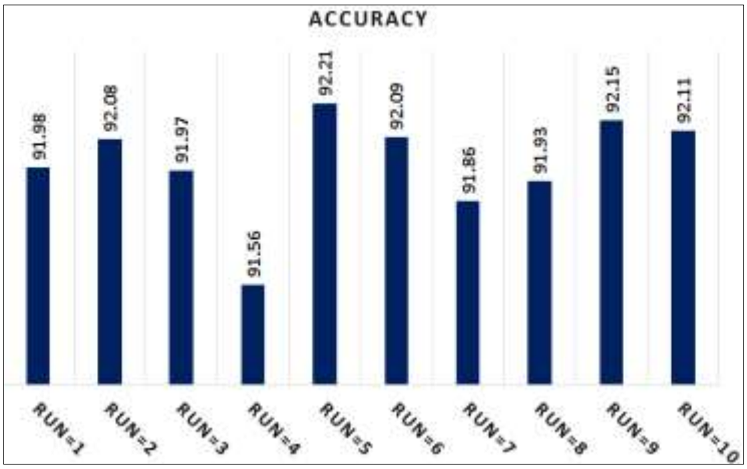

Fig. 7: Run 10 times for 200 iterations of the proposed model

4.3. Comparison and Evaluation

In Table (4), the comparison of the proposed model with various models is shown based on the accuracy criterion. In Table (4), the highest detection accuracy belongs to the FS model based on the Genetic Algorithm (GA). K-Nearest Neighbours (KNN) and SVM are more accurately compared with other classifications. The PSO with the KNN has the highest percentage of accuracy. The GA for the classification of multi-layer Artificial Neural Network (MLP ANN) is more reliable than the other classifications. The percentage of accuracy in the GA model with Decision Tree (DT) is 92.60%. The FS based on GA with the Boosting is 90.18%. The accuracy percentage in the FS model is based on the Feature Similarity with the KNN to 80.81%. The lowest percentage of accuracy belonging to the FS model is based on the Consistency with the MLP ANN classifier.

Table 4: Comparison of the proposed model with different models based on the percentage of accuracy

Model [17]	Classifier [17]						
	NB	SVM	KNN	Boosting	MLP ANN	Sequential Minimal Optimization (SMO)	DT
FS Feature Similarity	66.70	79.00	80.81	66.85	79.50	80.00	77.00
Laplacian Score for FS (LSFS)	69.30	83.80	82.68	69.28	80.60	79.30	69.10
Multi Cluster FS (MCFS)	65.30	80.00	82.27	65.25	69.30	73.30	72.60
Dense Subgraph Finding with Feature Clustering (DSFFC)	75.60	86.70	84.31	75.71	70.20	71.60	69.90
CFS	76.30	79.10	78.59	70.00	71.20	72.60	70.10
Consistency based FS (CON)	70.00	70.00	69.03	68.95	61.00	62.00	65.10
PSO	73.50	79.10	81.00	72.35	71.00	73.60	76.00
GA	70.20	62.10	63.39	69.99	72.50	70.30	70.10
FS-GA	80.90	91.50	92.22	90.18	92.00	88.00	92.60
Proposed Model	92.21						

Table 5: Comparison of proposed model with different models based on different criteria

Models [17]	Recall	Fallout	Feature	F1-Score
FS Feature Similarity	76.00	24.00	76.00	76.00
LSFS	76.00	24.00	76.00	76.00
MCFS	73.00	27.00	73.00	73.00
DSFFC	76.00	24.00	76.00	76.00
CFS	74.00	26.00	74.00	74.00
CON	66.00	33.00	67.00	67.00
PSO	75.00	25.00	75.00	75.00
GA	68.00	32.00	68.00	68.00
FS-GA	89.00	10.00	89.00	89.00
Proposed Model	90.49	8.12	91.88	91.00

Table 6: Comparison of the proposed model with different models based on the accuracy criterion

Models [18]						
Classifier [18]	GCNC	TV	LS	RRFS	ACO	All of FS
SVM	88.21	83.96	84.34	87.60	87.92	88.18
DT	89.05	83.03	85.61	85.71	86.97	88.81
NB	88.11	81.96	81.96	82.71	86.48	81.97
KNN	88.46	81.07	83.00	85.71	88.03	88.18
Proposed Model	92.21					

In Table (5), the comparison of the proposed model with other models is shown based on recall criteria, Fallout, features and F1-Score. The recall rate in the FS model based on the GA [17] is equal to 89.00%. The F1-Score and recall in the PSO [17] are 75% and 75%, respectively. The proposed model is more accurately compared to other models and has a lower error rate compared to other models that is equal to 8.12.

In Table (6), the comparison of proposed model with SVM classifier, DT, NB and KNN based on the accuracy criterion has been shown that the proposed model is better than the proposed methods [18]. The accuracy of Decision Tree (DT) with all features is more than the other classifications. ACO with KNN has better detection accuracy.

In Table (7), the comparison of the proposed model with different classifications based on the accuracy criterion is shown. In Table (7), the RELIEFF model is more accurate with C4.5, NB, KNN, and SVM. The NB is a Bayes theorem based statistical machine learning based method having properties of strong independence, probability distribution, and the ability to handle large datasets.

Table 7: Comparison of the proposed model with different models based on the accuracy

Models [19]	Classifier			
	C4.5	NB	KNN	SVM
CFS	81.16	57.69	79.14	85.85
	79.73	58.87	79.92	85.46
	79.73	58.87	79.92	85.46
	79.73	58.87	79.92	85.46
INT	78.16	57.95	79.73	80.31
	80.44	78.42	76.92	81.10
	80.05	61.73	76.79	81.88
	80.05	61.73	76.34	81.88
CONS	84.62	91.00	80.83	81.88
	85.27	92.05	76.79	81.16
	80.73	91.00	76.79	80.38
	80.83	91.00	76.79	80.38
IG	80.83	76.53	78.62	83.83
	81.66	66.95	77.71	83.38
	85.40	90.35	78.03	83.51
	85.40	90.35	78.03	83.51
RELIEFF	78.81	41.85	76.99	81.94
	84.88	92.05	80.18	83.77
	84.22	92.51	82.72	85.59
	84.22	92.50	82.72	85.60
Proposed Model	92.21			

Table 8: Comparison of proposed model with different models based on different criteria

Models	Accuracy					$MCC=\frac{(TP\times TN+FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$				
	Classifier [20]					Classifier [20]				
	SVM	NB	KNN	C4.5	Adaboost-NB	SVM	NB	KNN	C4.5	Adaboost-NB
UFSFS [20]	78.93	66.71	80.80	89.13	66.85	55.4	45.7	61.2	77.4	45.9
LSFS [20]	83.80	69.26	82.70	88.70	69.27	65.8	49.7	63.3	76.2	49.7
MCFS [20]	80.00	65.27	82.19	89.48	65.35	58.6	45.0	62.2	77.9	45.1
UDFS [20]	84.66	72.05	83.67	89.26	72.06	67.7	53.7	65.8	77.4	53.7
IMODEFS [20]	87.81	76.12	85.47	91.83	75.99	74.4	59.6	69.3	82.9	59.5
FSFS [21]	78.95	66.68	80.81	-	66.85	55.4	45.6	61.3	-	45.9
LSFS [21]	83.84	69.26	82.68	-	69.28	65.9	49.7	63.3	-	49.7
MCFS [21]	80.00	65.27	82.27	-	65.24	58.6	45.1	62.4	-	48.9
DSFFC [21]	86.69	75.63	84.31	-	75.71	71.9	58.5	66.8	-	58.6
Proposed Model	92.21					70.35				

Table 9: Comparison of proposed models with different models based on the accuracy

Classifier [22]	Models [22]							
	GCACO	L-Score	F-Score	RRFS	MRMR	RELIEFF	UFSACO	All of Features
SVM	88.38	83.96	86.55	87.71	87.51	86.14	78.92	88.81
DT	89.21	85.32	86.86	86.97	87.20	85.81	88.01	88.93
NB	88.22	81.96	86.62	83.04	80.50	85.50	86.48	83.05
KNN	88.94	83.09	86.41	85.40	84.45	85.62	85.16	88.62
RF	89.19	81.69	85.46	86.38	87.82	84.39	87.45	88.24
Proposed Model	92.21							

In Table (8), the comparison of the proposed model with different classifications is shown based on the MCC and accuracy criteria. In Table (8), the smallest MCC belongs to KNN. KNN, compared to other classifications has a higher accuracy of diagnosis and the lower amount of error. NB and Adaboost-NB classification are less accurate than other classifications. In Table (8), some models are applied in the unsupervised domain including Unsupervised FS using Feature Similarity (UFSFS), Laplacian Score for FS (LSFS), Multi-Cluster FS (MCFS), and Unsupervised Discriminative FS (UDFS) [20]. A modified model of DE called MODE has been proposed, where both local and global information are saved to make the convergence process faster as compared to the DE. Improved model of MODE (IMODE) based unsupervised FS (IMODEFS) has been proposed to search in the features [20]. An unsupervised FS algorithm has been developed by integrating the concept of densest subgraph with feature clustering (DSFFC) [21]. In (DSFFC), feature clustering around the non-redundant features is performed to produce the reduced feature set.

In Table (10), the comparison of the proposed model with various models is shown based on the accuracy criterion. The proposed model is more accurate than most models, such as GA, PSO, ACO, SVM, KNN, and C4.5. The differential column represents the difference in the accuracy of the diagnosis in the proposed model with other models.

In the proposed model, FFO is used to optimize the PSO. With the aid of particles, the similarity between the characteristics is measured. Then, the training and testing steps are carried out.

To classify features, a distance criterion based on FFO has been used. Also, in previous studies, a hybridization of algorithms for PSO and NSA and DE algorithms and NSA for email spam detection has been used. The proposed model is evaluated based on FS and various iterations. The accuracy value is greater with fewer features and 200 iterations. The results showed that the proposed model is more accurate in comparison with the FS based on similarity, FS based on GA, PSO, ACO, DE and statistical models of FS. Also, in other models, the NB, SVM, KNN, Boosting, and DT are used [18] in which ANN and KNN have better percentage of accuracy.

In Table (9), the comparison of the proposed model with the SVM classifier, DT, DT and KNN is shown based on the accuracy criterion.

5. Conclusions and Future Works

Although the email has many benefits, but one of its negative aspects is sending bulk spam to users. Usually organizations and individuals are involved with spam and are tired of removing them from their e-mail inbox. The main goal of spammers is to encourage users to open emails by sending various spam emails as they use emails sent from virus-infected files to spoil the web. Identification and classification are the most important factors to prevent spam. In this paper, a hybrid model based on PSO and FFO was used to email spam detection. Detection of specific features are most likely to be critical in email spam classification. This paper has applied a number of features in email spam which have resulted a different level of accuracy. To evaluate the proposed model, the spambase dataset was used and its results were compared with the meta-heuristic algorithms, machine learning, and DT. The results showed that the accuracy of the proposed model with all features is 92.21%, and the superiority of the proposed model is on average 25% compared to the comparative models. The accuracy result showed that the proposed model was competitive with the others methods. One of the most important weaknesses of the meta-heuristic algorithms is the proper adjustment of their parameters. These methods have weaknesses in local and global searches, which will result in proper adjustment of the parameters to reduce their runtime. Using this method will significantly increase the speed of convergence, the accuracy of finding the final answer, not being in the local points, and reducing the run-time.

Table 10: Comparison of the proposed model with various models based on the accuracy

Refs	Models	Accuracy
[5]	PSO+NSA	91.22
[6]	DE+NSA	65.14
[23]	LS	65.93
	RELIEFF	81.41
	MAXVAR	65.98
	MRMR	66.00
	MIM	74.84
	SDFS	72.98
	JSDFS	82.32
[24]	SFS	87.4
	SBS	87.01
	FAEMODE	89.48
	MOEA/DFS	88.48
[25]	GA	85.90
	EGA	86.24
	IGA	86.27
	BPSO	85.01
	BDE	86.53
	BACO	87.30
	ABACO	88.06
	GA-ACO	87.77
	PMBACO	88.47
	VMBACO	89.41
[26]	ABACO	92.30
	ABACO	92.10
	ACOFS	92.20
	BACO	91.90
	ACO	91.30
	ACO	90.10
	BGA	90.60
	BPSO	90.00
	IBGSA	92.20
	Catfish-BPSO	92.40
[27]	BPNN	89.70
	LVQ	89.80
	SVM	93.19
	1NN	81.32
	EM + 1NN	94.30
	C4.5	92.05
	RST	94.59
	SLDA	87.56
	GR + 1NN	90.77
	GA + 1NN	91.55
[28]	NB & FSS-MGSA	88.34
	ID3 & FSS-MGSA	77.24
[29]	NB-MICAP	74.30
	NB-IG	74.80
	NB-Relief	72.30
	NB-RFE	75.70
[30]	EIS-RFS	89.35
	IS-SSGA	82.60
	FS-SSGA	83.47
	IFS-SSGA	87.54
	FS-RST	81.74
	FS-RST + IS-SSGA	76.93
	IS-SSGA + FS-RST	79.43
	1-NN	77.89
-	Proposed Model	92.21

The following items can be mentioned as future works:

- Improve the proposed model in terms of classification accuracy
- Combine data mining methods and meta-heuristic algorithms to select important features and increase the accuracy of classification
- Use the fuzzy inference system to select important features
- Test the proposed model with real data

References

[1] C. Meli, V. Nezval, Z.K. Oplatkova, V. Buttigieg, “Spam detection using linear genetic programming,” 23rd International Conference on Soft Computing, MENDEL 2017: Recent Advances in Soft Computing, Vol. 837, pp. 80-92, (2019).

[2] S. Saha, S. Das, G. Suman, K. Das, “Spam mail detection using data mining: A comparative analysis,” *Smart Intelligent Computing and Applications*, vol. 104, pp. 571-580, (2019).

[3] F. S. Gharehchopogh, H. Gholizadeh, “A comprehensive survey: Whale Optimization Algorithm and its applications,” *Swarm and Evolutionary Computation*, Vol. 48, 1-24, (2019).

[4] S. B. Rathod, T. M. Pattewar, “Content based spam detection in email using Bayesian classifier”, *International Conference on Communications and Signal Processing (ICCSP)*, pp. 1257-1261, (2015).

[5] I. Idris, A. Selamat, “Improved email spam detection model with negative selection algorithm and particle swarm optimization”, *Applied Soft Computing*, Vol. 22 pp. 11-27, (2014).

[6] Idris, A. Selamat, S. Omatu, “Hybrid email spam detection model with negative selection algorithm and differential evolution”, *Engineering Applications of Artificial Intelligence*, Vol. 28, 97-110, (2014).

[7] R. Chikh, S. Chikhi, “Clustered negative selection algorithm and fruit fly optimization for email spam detection,” Journal of Ambient Intelligence and Humanized Computing, Vol. 10, pp. 143-152, (2017).

[8] S. A. Khamis, C. F. M. Foozy, M. F. A. Aziz, N. Rahim, “Header based email spam detection framework using Support Vector Machine (SVM) Technique”, *International Conference on Soft Computing and Data Mining SCDM 2020: Recent Advances on Soft Computing and Data Mining*, Vol. 978, pp. 57-65, (2020).

[9] S. O. Olatunji, “Improved email spam detection model based on support vector machines,” *Neural Computing and Applications*, Vol. 31, 691-699, (2019

[10] W. T. Pan, “A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example”, *Knowledge-Based Systems*, Vol. 26, pp. 69-74, (2012).

[11] J. Kennedy, R. C. Eberhart, “Particle Swarm Optimization”, In Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, (1995).

[12] M. Mahmoudi, F. S. Gharehchopogh, “An improvement of shuffled frog leaping algorithm with

a decision tree for feature selection in text document classification," *CSI Journal on Computer Science and Engineering,* Vol. 16, No. 1, pp. 60-72, (2018).

[13] A. Allahverdipour, F. S. Gharehchopogh, "An Improved K-Nearest Neighbor with Crow Search Algorithm for Feature Selection in Text Documents Classification," *Journal of Advances in Computer Research*, Vol. 9, No. 2, pp. 37-48, (2018).

[14] H. Majidpour, F. S. Gharehchopogh, "An improved flower pollination algorithm with adaboost algorithm for feature selection in text documents classification," *Journal of Advances in Computer Research,* Vol. 9, No. 1, pp. 29-40, (2018).

[15] S. Ardam, F. S. Gharehchopogh, "Diagnosing liver disease using firefly algorithm based on Adaboost," *Journal of Health Administration,* Vol. 22, No. 1, pp. 61-77, (2019).

[16] Spambase Data set: Available online at: https://archive.ics.uci.edu/ml/datasets/spambase, [last available: 2019.02.05]

[17] A. K. Das, S. Das, A. Ghosh, "Ensemble feature selection using bi-objective genetic algorithm", *Knowledge-Based Systems*, Vol. 123, pp. 116-127, (2017).

[18] P. Moradi, M. Rostami, "A graph theoretic approach for unsupervised feature selection," *Engineering Applications of Artificial Intelligence,* Vol. 44, pp. 33-45, (2014).

[19] L. M. Fernandez, V. B. Canedo, A. A. Betanzos, "Centralized vs. distributed feature selection methods based on data complexity measures", *Knowledge-Based Systems,* Vol. 117, pp. 27-45, (2017).

[20] T. Bhadra, S. Bandyopadhyay, "Unsupervised feature selection using an improved version of differential evolution", *Expert Systems with Applications*, Vol. 42, No. 8, pp. 4042-4053, (2015).

[21] S. Bandyopadhyay, T. Bhadra, P. Mitra, U. Maulik, "Integration of dense subgraph finding with feature clustering for unsupervised feature selection", *Pattern Recognition Letters,* Vol. pp. 104-112, (2014).

[22] P. Moradi, M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection", *Knowledge-Based Systems,* Vol. 84, pp. 144-161, (2015).

[23] H. Yan, J. Yang, "Sparse discriminative feature selection," *Pattern Recognition*, vol. 48, pp. 1827–1835, (2015).

[24] S. K. Nayak, P. K. Rout, A. K. Jagadev, T. Swarnkar, "Elitism based multi-objective differential evolution for feature selection: A filter approach with an efficient redundancy measure", *Journal of King Saud University-Computer and Information Sciences,* Vol. 32, No. 2, pp. 174-187, (2017).

[25] Y. Wan, M. Wang, Z. Ye, X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm", *Applied Soft Computing,* Vol. 49, pp. 248-258, (2016).

[26] S. Kashef, H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection", *Neurocomputing,* Vol. 147, pp. 271-279, (2015).

[27] C. T. Su, H. C. Lin, "Applying electromagnetism-like mechanism for Feature Selection", Information Sciences, Vol. 181, pp. 972-986, (2011).

[28] X. H. Han, X. M. Chang, L. Quan, X. Y. Xiong, J. X. Li, Z. X. Zhang, Y. Liu, "Feature subset selection by gravitational search algorithm optimization", *Information Sciences,* Vol. 281, pp. 128-146, (2014.

[29] X. Zhao, W. Deng, Y. Shi, "Feature selection with attributes clustering by maximal information coefficient", *Procedia Computer Science,* Vol. 17, pp. 70-79, (2013).

[30] J. Derrac, C. Cornelis, S. Garcia, F. Herrera, "Enhancing evolutionary instance selection algorithms by means of fuzzy rough set-based feature selection", *Information Sciences,* Vol. 186, No. 1, pp. 73-92, (2012).