ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 9, No. 3, August 2021, pp.520-531

An Efficient feature selection algorithm for the spam email classification

Hadeel M. Saleh
Continuing Education Center, University of Anbar, Iraq

ABSTRACT

The existing spam email classification systems are suffering from the problems of low accuracy due to the high dimensionality of the associated feature selection (FS) process. But being a global optimization process in machine learning, FS is mainly aimed at reducing the redundancy of dataset to create a set of acceptable and accurate results. This study presents the combination of Chaotic Particle Swarm Optimization (PSO) algorithm with Artificial Bees Colony (ABC) for the reduction of features dimensionality in a bid to improve spam emails classification accuracy. The features for each particle in this work were represented in a binary form, meaning that they were transformed into binary using a sigmoid function. The features selection was based on a fitness function that depended on the obtained accuracy using SVM. The proposed system was evaluated for performance by considering the performance of the classifier and the selected features vectors dimension which served as the input to the classifier; this evaluation was done using the Spam Base dataset and from the results, the PSO-ABC classifier performed well in terms of FS even with a small set of selected features.

Keywords: Feature Selection, Hybrid Algorithm, Swarm Intelligence, Machine Learning Classification, Spam Filtering

Corresponding Author:

Hadeel M. Saleh
Continuing Education Center, University of Anbar
Ramadi, Iraq
Email: haddeel.mohammed@uoanbar.edu.iq

1. Introduction

E-mails are generally considered a reliable channel of communication and as such, has recently become the target of numerous attacks. A common form of these attacks is junk or spam emails; these junk emails are deliberately delivered to the target using different protocols, like SMTP [1][2]. They are sent in high numbers and as such, occupies a significant portion of network bandwidth. Spam emails can also be annoying and can deprive users of using the available network resources because they compete with the legal users for the available storage space on the server. Spam emails also causes wastage of precious communication effort and time; they are also a source of threat to official establishments [3] [4]. The detection of spam emails is generally done by appropriately classifying incoming emails into spam & non-spam classes. Most new spam detection systems are ML-based [5] [6] but one of the common problems encountered is how to select the optimal input feature subsets for the selected classifiers. This is normally done via FS processes and this is usually hampered by the issue of high data dimensionality associated with the FS process as it reduces the performance of some classifiers, like SVM, ANN, and NBC [7]–[15]. This high data dimensionality can be prevented by reducing the feature space; this can be achieved by minimizing the number of features present in the data. But it is proper to ensure that the FS process returned features that will represent the problems encountered in the document. Irrelevant features can impact the classification accuracy as well and can affect the time needed to train the classifier; it can also affect the feature-related expenditure and the number of required instances for learning [16], [17].

The evolutionary and swarm-based techniques, such as ACO [18], [19], GA [20], [21], ABC [21] [22], PSO [23] [24], and HSA have been the commonly used methods for addressing FS-related problems [26], [27]. As a nature-inspired framework, PSO [24], [28], [29] was developed based on inspiration from the natural way of life of fish and birds; it has been used in finding solution to different complex optimization tasks. Since its introduction by [28], PSO has been modified severally, giving rise to several version of the algorithm; the aim of these modifications is to find a better way of addressing specific optimization tasks. The modifications on the PSO variants are in different categories as follows: (i) modification focused on the parameter settings with more attention on the optimization of the acceleration and inertia weight coefficient parameters; (ii) modifications based on the topology of the neighborhood that portrays the connection between the particles; (iii) those that consider the learning techniques; (iv) those that deals mainly on the combination of PSO variants with other algorithms [30]–[36].

A wrapper FS method based on tent chaotic map and binary PSO-BABC is proposed in this study for fitness evaluation. The role of the suggested method is to ensure the selection of the best subset features from the Spam base dataset to ensure better classification and filtering of junk emails. The remaining parts of this article are arranged as follows: Section 2 presents the description of the standard PSO and ABC; Section 3 detailed the proposed approach; Section 4 presents the results of the evaluations; Section 5 concluded the study.

2. Overview

2.1 Optimization problems and algorithms

Several optimization problems are encountered in every field of study and the problem is that there is no known solution to these problems as no specific method or algorithm has been found to provide the optimal solution within a specific period. Hence, most optimization problems are being referred to as Non-Detereminstic Polynomial Time (NP) and could be generally classified into constrained and unconstrained problems based on the existance of constraints. These problems are generally formulated mathematically as follows:

$$Min/Max \;\; f_i(X) = \{i = 1,2,3 \ldots D\} \tag{1}$$

S.T

$$g_k(x) \leq 0 \;\; \{k = 1,2,3 \ldots, K\} \tag{2}$$

$$h_j(x) = 0 \quad \{j = 1,2,3, \ldots, J\} \tag{3}$$

where $f_i(x)$ is the objective function of the problem to be either maximized or minimized, $g_k(x)$ and $h_j(x)$ are the inequality constraint and equality constraint of the problem, while x is a specific decision variable, or a specific vector of the decision variables.

The problem itself can be affected by the number of decision variables; there are 2 types of problems-small/normal scale problems (number of decision variables (DVs) is <100) and large-scale problems (the number of DVs is > 100). Sometimes, these problems may have thousands of DVs and in such cases, it is hard to execute the search process.

Numerous optimizers have been proposed and developed in the past few decades with the aim of finding solution to these optimization problems. Most of these optimizers are nature-inspired as they mimic natural processes and social behaviors (such as the movement of birds, light flashing pattern of fireflies, grey wolf hunting strategy, movement of normads, etc.) [7]. Some optimizers are also based on mathematical formulations, such as Scine-Cosin Algorithm (SCA) which is used to solve the issue of Solid Waste Collection in this work. Few optimizers are also based on physical process, such as the BHA which copies the behavior of the black hole in the universe, as well as the Multiverse Optimization Algorithm (MVO) that mimic the multi-universe theory [6], [8]–[10]. Metaheuristics has been successfully adopted in many fields and their success in these fields has attracted more attention for using them in solving optimization tasks. Researchers in ML have resorted to various metaheuristics for the improvement of the performance of their prediction models [11]–[14]. Metaheuristics are also used to solve engineering problems [15], [16]. Efforts have also been made towards using metaheuristics to solve FS problem, which is a known optimization problem in data science [17], [18], as well as other applications that abound in the literature [19]–[28]. The two major components of any population-based algorithm are the exploration and explotation capabilities of the algorithm; both components must be balanced based on the specific optimization problem. This is important because increasing in one component will slow the other component and could cause local optima entrapment [29], [30].

2.2 Artificial bees colony (ABC)

The ABC was developed by [37], [38] as an algorithm with 3 major groups of bees (employed bees (EB), onlookers (OLB), and scout bees(SB)) which work together to find new food foods; the EBs are responsible for finding the location of new food sources and sending the signal to the OLBs. The OLBs receive the information from the EBs in the form of a waggle dance in the hive. The nature of dance is directly proportional to the nectar content of new food source found by the EBs. The OLBs select the potential food source based on the intensity of the dance; more onlookers are attracted to the good food sources compared to the bad ones. Here, the food source is doing the exploration job while the EBs and OLBs are doing the exploitation job.

The ABC considers each food source as a potential solution to the considered problem while the nectar content of the food source represents the quality of the solution as represented by the fitness value. For each food source, only one employed bee is selected as the number of food sources is equivalent to the number of employed bees. The food source selection by the onlooker bee is based on the probability value P_i that is related to the food source:

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{1}$$

where fit_i represent the fitness value of the solution i; SN is the total number of food sources. ABC uses the following expression to produce a specific food position $V_i = \{v_{i,1}, v_{i,2}, \dots., v_{i,D}\}$ from the old one $X_i = \{X_{i,1}, X_{i,2}, \dots, X_{i,D}\}$ in memory:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{2}$$

“Where $k \in \{1,2, \dots, SN\}$ and $j \in \{1,2, \dots, D\}$ are indexes that are selected randomly; k must differ from i; D is the overall number of variables; ϕ is a random number ranging from $[-1,1]$.”

After the artificial bee has produced and evaluated the position of each candidate food source, the performance of the evaluated source will be matched with that of the old one and if the quality of the new source is better or equal to the old source, the new one will replace the old one, else, the old one will be retained. If there is no further chance of improving a problem by a predefined number of cycles, then, it is assumed that that food source has been abandoned. The number of predetermined cycles is an important ABC control parameter and represntts the limit for abandonment. Consider the abandoned source as X_i and $j \in \{1, 2, \dots, D\}$, then, the scout identifies a new source of food to be replaced with X_i. This process can be defined thus:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}) \tag{3}$$

2.3 Particle swarm optimization (PSO)

PSO was presented as a nature-inspired metaheuristics that mimics the flocking pattern of birds during active search for food. The PSO considers a random distribution of the flock of birds with only one source of food as seen in Fig 2 (the only food source is the dot on the tree). Despite placing those food source at a known distance to each bird, the birds are not aware of its position but the nearest bird to the food piece can communicate with the distal birds to facilitate flocking towards the source of food. Hence, each bird in the swarm is seen as a particle while the source of food is the optimal value. The value of the objective function represents the distance of the food source from each bird; hence, this flocking behavior can be regarded as a function optimization problem. The most proximal bird to the food piece is denoted as X_i (Figure 1) and is the existing best, while its distance from the best position is represented as N_{best_i} [28], [39].

Each particle in the PSO is assumed to have a specific velocity and position during an active search for optimal solution. Hence, each particles' position can be improved based on its current local & global best; this can be done, say for particle i, as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

where t = current status, $t+1$ = status after update, X_i(t+1) = new particles' velocity.
Observe that the time variation $\Delta t = (t+1) - 1$ is the time unit while particle i's velocity is given as:

$$V_i(i+t) = \omega V_i(t) + c_i r_i \left(X_i^P - X_i(t)\right) + c_2 r_2 \left(X^G - X_i(t)\right) \quad (5)$$

where $v_i(t)$ = current particle velocity, X_i^P = swarm's global best, & X^G = particle's global best, ω, c_1, & c_2 = constants that determine each velocity component's relevance, r_1 & r_2 = random values range from 0-1.

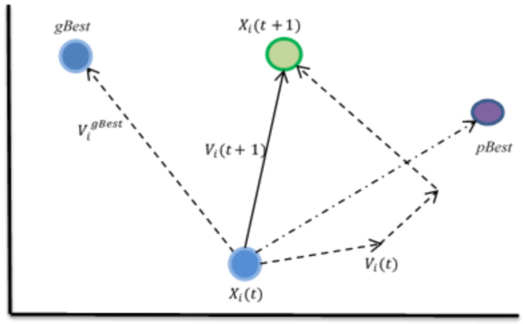

Figure 1. A depiction of the PSO algorithm

PSO has undergone several modifications, yet, it still has various issues, such as premature convergence, low convergence speed, local minima entrapment, inbuilt complexity, multimodality, and discontinuity problems which often causes low solution quality, and solution uncertainty [33], [34].

3. The proposed algorithm

In general, the feature selection stage is implemented within the preprocessing phase, meaning that the most relevant subset of features should be determined before the classification phase. However, the wrapper type of feature selection methods requires a classifier model for evaluating the fitness of each generated/updated solutions. The normalization stage is performed before the feature selection stage. In this study, $MinMax$ normalization method is executed.

FS algorithms are developed to find better subset features that could guarantee improved performance accuracies. The particles in the proposed system were initialized from random positions that are generated using tent map; these positions are further transformed into binary. The dataset is first read and normalized before executing the algorithm. Figure 2 depicts the flowchart of the suggested algorithm, showing the six major steps of the algorithm as follows:

Step1: Generation of the initial particles and bees position using the chaotic tent map equation as follows:

$$X_{i+1} = \begin{cases} \frac{x_i}{0.7} & if\ x_i < 0.7 \\ \frac{10}{3}(1-x_i) & otherwise \end{cases} \quad (6)$$

where X_i= a real value in the range of 0 and 1 and represents one dimension of any given problem.

Step2: Conversion of all Xi values into binary using the sigmoid function as follows:

$$F_i = \begin{cases} 1, sigmoid\ (X_i) > u\ [0,1] \\ 0, \quad otherwise \end{cases} \quad (7)$$

where: X_i = each particle's position, $sigmoid(X_i) = 1 / [1 + e^{-X}]$, u = uniform distribution, F_i = binary sequence, 1 = the propability of choosing a feature, 0 = probability of not choosing a feature.

Step3: Calculation of each particles' fitness function using their binary sequence. In the proposed case study, the objective function is the obtained classification accuracy using SVM. Hence, the proposed scheme in this study strives towards maximization of the fitness function, that is, maximization of the classification accuracy of SVM via the selection of the most minimum relevant features. The following equation is used to calculate the fitness value:

$$\min err = 1 - \frac{C.Acc}{O.Acc} + (a \times \frac{C.F}{O.F}) \qquad (8)$$

where $C.Acc$ and $O.Acc$ represent the accuracy of the current solution with selected $C.F$ features and the original accuracy based on all features, i.e., $O.F$, and a represents a real number which is in range [0,1].

Step4: The generated solutions in the population should be updated and enhanced, however, the updating procedure is done based on the PSO or ABC algorithm. If a uniform distribution number is less than 0.5, then updated the solutions based on ABC algorithm, otherwise, update the solutions based on PSO.

For ABC:

- Use Eq. 2 to update the employed bees (the local search aspect); then, select one employed bee in consideration of its probability using Eq.1. Replace any abandoned abandoned solution for the scout with a new solution that is randomly generated using Eq. 3.

For PSO:

- Update each particles' position and velocity using Eqs. 4 & 5. After the update, convert the resulting positions into binary as in step 2. Then, calculate the fitness function again as in step 3. The last step process is updating of the local and global best solutions.

Step5: The best solution in terms of the accuracy is determined and kept for the next iteration, or returned as the final solution.

Step6: Check for the termination condition; it met, stop the algorithm and return the gbest; else, revert to step 4.

The flowchart of the proposed algorithm is presented in the following figure.

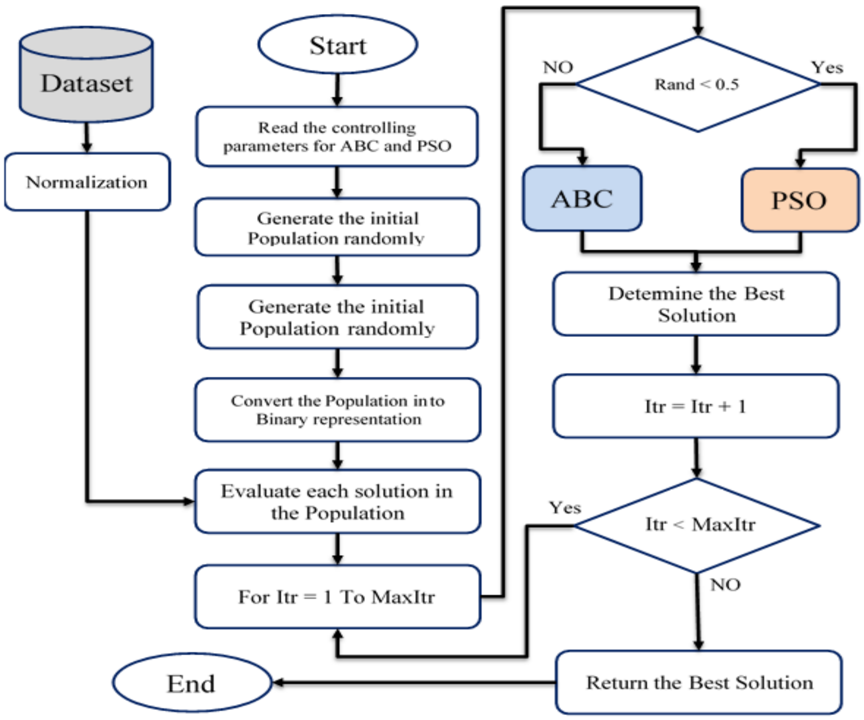

Figure 2. The flowchart of the proposed algorithm

4. Data description

The spam filtering system is usually classifying the emails into two main classes: spam (1) or in some studies called "ham", and normal or not-spam (0). Mainly, there are two types of classification procedures, first, based on the content of the emails, i.e., the text. While the second type is based on the sender information such as the IP-address, the port number, or the server information. The dataset used in this study is conducted based on the contents of the emails, therefore, it belongs to the first type. The dataset is called “SpamBase”, which consists of 4601 emails, all these emails have been processed and a set of 57 features were extracted. Most of these 57 features are the most common words in the spam emails, such as the word “money”, or “price”. While some of these features are the characteristics of the word or characters, such as the number of capital letters in the message, or the number of symbols. The last two features measure the length of the succession of back-to-back capital letters.

The SPAMBASE dataset was downloaded from the UCI ML repository [40]. In this dataset, the non-spam emails were pooled from field works, personal e-mails, and single e-mail accounts. These emails in the dataset are considered suitable for the evaluation of the performance of new spam filtering techniques. Each instance in the SPAMBASE contains 58 attributes, of which most are the frequency of a given email character that corresponds to the instance.

5. Results and discussion

In this study, there are two experiments for evaluating the proposed hybrid algorithm. In first experiment, the ability for handling the optimization problems using ABC-PSO algorithm is evaluated. While in the second experiment, the ABC-PSO algorithm is utilized for solving the problem of selecting the best subset of features.

5.1 ABC-PSO for Numerical Optimization Problems

Four numerical optimization problems were used in the experiments as presented in the following: The two major parameters of the ABC-PSO are i) the swarm size (SS) (represents the size of the solutions in the swarm), and ii) the (Itr) which is the number of runs. In this study, SS = 50, while $Itr = \{100{,}500\}$. The results of the proposed algorithm based on the above-mentioned test functions are given in Table 2 below. The comparison of the new hybrid method with the standard version of the algorithm shows that our algorithm attained a superior performance, and was more stable than the algorithm based on the standard deviation (S.D). It can be seen that our algorithm has the ability to perfume a better local search based on the first and the second test function, because of the combination between the position updating mechanism of PSO algorithm, and the onlooker bees. Meaning that, there is a high possibility for escaping the local minima. On the other hand, the combination between the updating velocity of PSO – especially the third part of eq. 5 – and the scout searching of ABC algorithm, helps to explore the search space for better positions, and also decrease the chance for getting trapped in the local minima. This ability has been proven based on the results of the last two test functions. I, the proposed hybrid algorithm has managed to handle different types of optimization problems, in terms of modality (i.e., unimodal vs. multimodal) and in terms of separability (i.e., separable vs. non-separable). Figure 3-6 below illustrates the convergence of the developed scheme, as matched to the standard ABC and PSO. It is obvious that ABC-PSO algorithm has converged towards the optimal solution faster than the original versions of the algorithms, meaning that it has better balancing between the local and global searching capabilities.

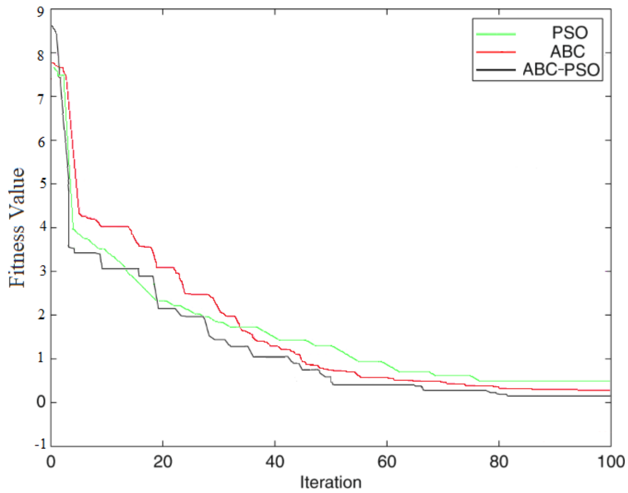

Figure 3. Convergence of F1

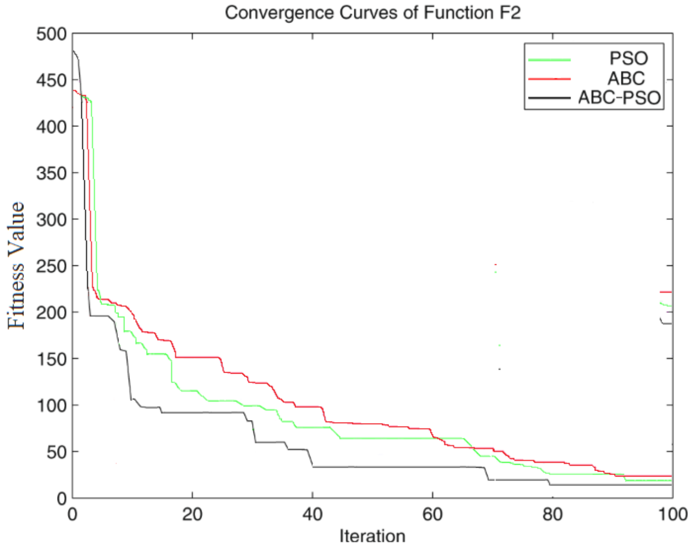

Figure 4. Convergence of F2

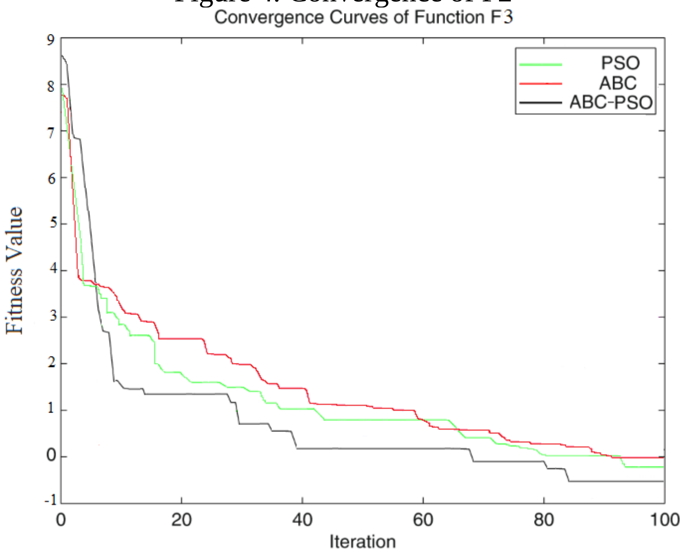

Figure 5. Convergence of F3

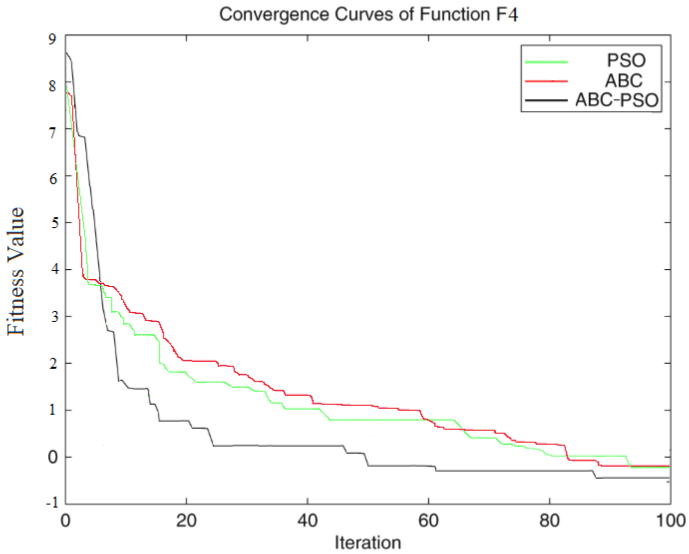

Figure 6. Convergence of F4

Table 2. The obtained results

Itr	f_n	Algorithm	Best	Mean	S.D
100	f_1	PSO	2.5457521	2.7647845	0.0784516
		ABC	0.9854126	1.0154784	0.0014784
		ABC-PSO	**0.0005784**	**0.0007741**	**0.0000106**
	f_2	PSO	0.0884741	0.0964587	0.0078478
		ABC	0.0078414	0.0087789	0.0009874
		ABC-PSO	**0.0000564**	**0.0000845**	**0.0000621**
	f_3	PSO	21.695847	27.947512	0.0847896
		ABC	2.0018977	2.6647845	0.0078487
		ABC-PSO	**0.0003850**	**0.0040184**	**0.0000945**
	f_4	PSO	16.4875218	26.110161	0.0238484
		ABC	1.99847	2.5869124	0.0084578
		ABC-PSO	**0.0002645**	**0.0014213**	**0.0000315**
500	f_1	PSO	7.2456571	2.7647845	0.0784516
		ABC	2.4859157	1.0154784	0.0014784
		ABC-PSO	**0.0021645**	**0.0007945**	**0.0000021**
	f_2	PSO	1.2785781	0.0964587	0.0078478
		ABC	0.9045472	0.0087789	0.0009874
		ABC-PSO	**0.0041443**	**0.0000487**	**0.0000584**
	f_3	PSO	48.995751	27.947512	0.0847896
		ABC	7.2214945	2.6647845	0.0078487
		ABC-PSO	**0.0706241**	**0.0040012**	**0.0000115**
	f_4	PSO	37.125475	26.110161	0.0238484
		ABC	3.35847	2.5869124	0.0084578
		ABC-PSO	**0.1084123**	**0.0014871**	**0.0000484**

5.2 ABC-PSO for E-mail spam filtering

Some evaluation metrics were used for the evaluation of the performance of the developed ABC-PSO on the considered dataset. Filtering accuracy is the simplest evaluation metric as it measures the percentage of instances that are classified correctly [41]. The accuracy represents the percentage of correctly classified emails into the right classes; it is determined using Eq. 5. The dataset was partitioned into 70% and 30% for the training & testing phases, respectively. The ABC-PSO was implemented for 20 runtimes using different sizes of swarms and iteration numbers for the sake of performance comparison in finding the best features subsets. The evaluation was done on a PC with the following specification: RAM size = 8 GB, 2.6 GHz core i7 of CPU. The MATLAB programing language was used to write and execute the algorithm. Two swarm sizes were considered (10 and 50), and each swarm size was tested for different iteration numbers (100, 200, and 300). The results of the experiments (in terms of best, worst, mean accuracy, mean selected features, and standard deviation) are presented in Table 3.

In general, the proposed algorithm showed a huge improvement on the original accuracy; ABC-PSO helped NBC by selecting the most relevant features. The original accuracy for NBC based on all 57 features was around 79.41%, while the worst result achieved by ABC-PSO was around 90%. The obtained results in more details are illustrated in Figure 8 and Figure 9, where each experiment has been executed for 10 run times.

Table 3. Results of the proposed algorithm

Iterations	Swarm Size	Best	Worst	Mean Accuracy	Mean Features	St.div
100	10	92.41	89.82	90.81	33.5	0.8121
	50	93.052	92.23	92.929	29.4	0.9218
200	10	93.16	90.014	91.926	31.2	1.0818
	50	94.25	93.478	93.386	26.2	0.8732
300	10	93.183	90.053	91.557	34.4	1.0587
	50	95.921	93.828	93.487	22.8	0.8531

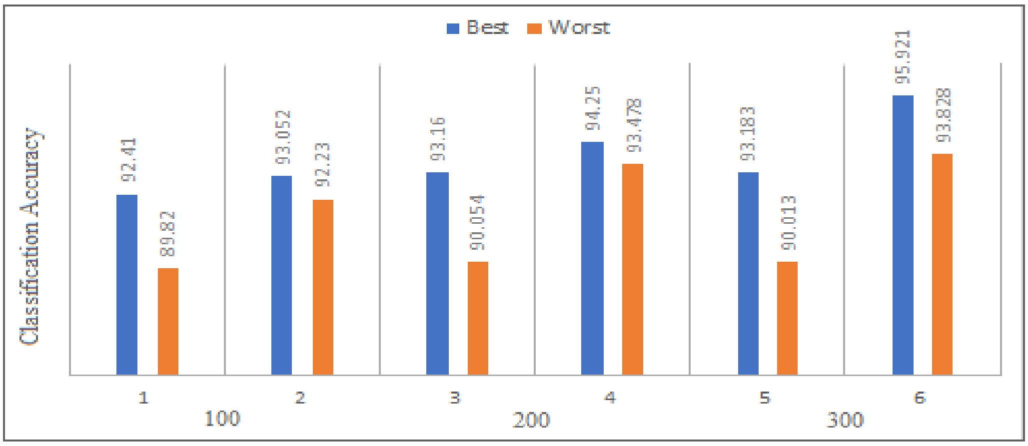

Figure 7. The classification accuracy of the proposed algorithm based on different swarm sizes and number of iterations

These figures illustrate the impact of the swarm sizes on the search performance of the proposed algorithm. The best results obtained so far are increased when the swarm size is increased. Moreover, the number of iterations has another impact on the algorithm, where the chances are increased for finding better solutions, meaning that both the swarm size and the number of iterations impacted the search process. However, in the worst cases, the algorithm performed well (average accuracy = 90.81; average number of selected features = 33.5).

In addition to the previous presented results, a comparison between the proposed algorithm and several recent algorithms was presented in Table 4. These algorithms are divided into two types, first, only classification models based on all features set (i.e., Features = 57), while the second algorithms are nature-inspired algorithms used for selecting the most relevant subset of features. The algorithms used for the comparison are SVM, KNN, ACO, GA, PSO, NSA, and DFS algorithms. Our proposed algorithm performed better than these other models and algorithms in all aspects.

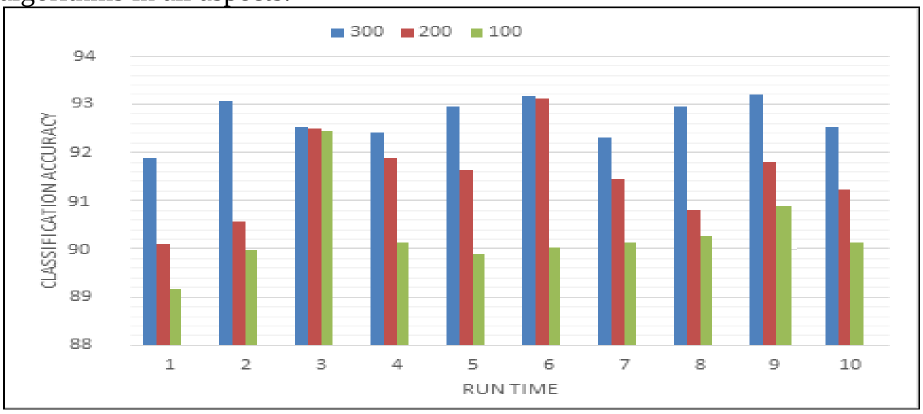

Figure 8. Results obtained using S.S = 10

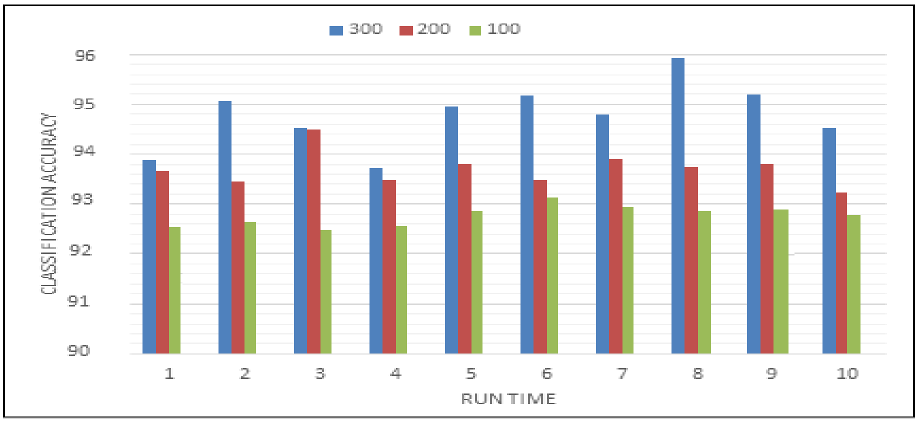

Figure 9. Results obtained using S.S = 50

Table 4. The comparison between ABC-PSO and other state of arts algorithms

Method	Classifier	Accuracy	Error	Ref
-	NBC	79.6	20.4	-
-	SVM	90.42	9.58	-
-	KNN	89.52	10.48	-
ACO	SVM	81.25	18.75	[42]
GA	NBC	77	23	[43]
ACO	NBC	84	16	[43]
PSO	NSA	82.62	17.38	[44]
DFS	SVM	71	29	[45]
ABC-PSO	**NBC**	**91.26**	**8.74**	**Proposed**

6. Conclusion

There are several types of anti-spam filters which are designed to manually filter e-mails. However, these methods require time and experience to work efficiently and requires a constant update of the features of all unwanted messages on a regular basis. Meanwhile, the automatic spam filtering systems are more beneficial than the manual method. Some ML methods have found application in text classification into categories based on their content. The effectiveness of spam filter models has been reported to rely on the improvement of the recognition performance of the classifiers and on the retraining of the benchmark models. Another important issue in spam filtering is the selection of the features which involves the selection of the sub-feature that will capture the whole information in the dataset. This study designed a hybrid metaheuristic for the odentification of the most relevant feature subset in SPAMBASE dataset; it is a hybrid metaheuristic that combined PSO and ABC. The proposed algorithm which is called "ABC-PSO" was initialized using a chaotic tent map method, which generates better position for explore search space than the original uniform distribution method. The results showed better performance of the proposed ABC-PSO than the other state of arts algorithms in the global optimization test functions, and in selecting the subset of features. For future studies, the proposed algorithm could be used for different optimization problems and feature selection case studies.

References

[1] D. Debarr and H. Wechsler, "Spam detection using Random Boost," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1237–1244, 2012.

[2] Q. Wu, S. Wu, and J. Liu, "Hybrid model based on SVM with Gaussian loss function and adaptive Gaussian PSO," *Eng. Appl. Artif. Intell.*, vol. 23, no. 4, pp. 487–494, 2010.

[3] S. M. Lee, D. S. Kim, J. H. Kim, and J. S. Park, "Spam Detection Using Feature Selection and Parameters Optimization," in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, 2010, pp. 883–888.

[4] N. Jindal and B. Liu, "Analyzing and detecting review spam," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2007, pp. 547–552.

[5] M. Hall, "Correlation-based Feature Selection for Machine Learning," *Methodology*, vol. 21i195-i20, no. April, pp. 1–5, 1999.

[6] M. Chang and C. K. Poon, "Using phrases as features in email classification," *J. Syst. Softw.*, vol. 82, no. 6, pp. 1036–1045, 2009.

[7] H. Liu, X. Shi, D. Guo, Z. Zhao, and Yimin, "Feature selection combined with neural network structure optimization for HIV-1 protease cleavage site prediction," *Biomed Res. Int.*, vol. 2015, 2015.

[8] S. Li, P. Wang, and L. Goel, "Wind Power Forecasting Using Neural Network Ensembles with Feature Selection," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1447–1456, 2015.

[9] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5197–5204, 2011.

[10] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 5432–5435, 2009.

[11] M. L. Zhang, J. M. Peña, and V. Robles, “Feature selection for multi-label naive Bayes classification,” *Inf. Sci. (Ny).*, vol. 179, no. 19, pp. 3218–3229, 2009.

[12] G. Feng, J. Guo, B.-Y. Jing, and T. Sun, “Feature subset selection using naive Bayes for text classification,” *Pattern Recognit. Lett.*, vol. 65, pp. 109–115, 2015.

[13] S. Q. Salih, “A New Training Method Based on Black Hole Algorithm for Convolutional Neural Network,” *J. Sourthwest Jiaotong Univ.*, vol. 54, no. 3, pp. 1–10, 2019.

[14] S. I. Abba, S. J. Hadi, S. S. Sammen, S. Q. Salih, R. A. Abdulkadir, Q. B. Pham, and Z. M. Yaseen, “Evolutionary computational intelligence algorithm coupled with self-tuning predictive model for water quality index determination,” J. Hydrol., vol. 587, p. 124974, Aug. 2020.

[15] Z. M. Yaseen, Z. H. Ali, S. Q. Salih, and N. Al-Ansari, “Prediction of Risk Delay in Construction Projects Using a Hybrid Artificial Intelligence Model,” Sustainability, vol. 12, no. 4, p. 1514, Feb. 2020.

[16] A. Unler, A. Murat, and R. B. Chinnam, “Mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification,” *Inf. Sci. (Ny).*, vol. 181, no. 20, pp. 4625–4641, 2011.

[17] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[18] P. Moradi and M. Rostami, “Integration of graph clustering with ant colony optimization for feature selection,” *Knowledge-Based Syst.*, vol. 84, pp. 144–161, 2015.

[19] S. M. Vieira, J. M. C. Sousa, and T. A. Runkler, “Two cooperative ant colonies for feature selection using fuzzy models,” *Expert Syst. Appl.*, vol. 37, no. 4, pp. 2714–2723, 2010.

[20] M. M. Kabir, M. Shahjahan, and K. Murase, “A new local search based hybrid genetic algorithm for feature selection,” *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, 2011.

[21] C.-H. Lin, H.-Y. Chen, and Y.-S. Wu, “Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection,” *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6611–6621, 2014.

[22] M. Schiezaro and H. Pedrini, “Data feature selection based on Artificial Bee Colony algorithm,” *J. Image Video Process.*, vol. 1, no. 47, pp. 1–8, 2013.

[23] V. Agrawal and S. Chandra, “Feature Selection using Artificial Bee Colony Algorithm for Medical Image Classification,” *2015 Eighth Int. Conf. Contemp. Comput.*, vol. 1, pp. 2–7, 2015.

[24] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms,” *Appl. Soft Comput. J.*, vol. 18, pp. 261–276, 2014.

[25] B. Xue, M. Zhang, S. Member, and W. N. Browne, “Particle Swarm Optimization for Feature Selection in Classification : A Multi-Objective Approach,” pp. 1–16, 2012.

[26] C. C. O. Ramos, A. N. Souza, G. Chiachia, A. X. Falc??o, and J. P. Papa, “A novel algorithm for feature selection using Harmony Search and its application for non-technical losses detection,” *Comput. Electr. Eng.*, vol. 37, no. 6, pp. 886–894, 2011.

[27] H. H. Inbarani, M. Bagyamathi, and A. T. Azar, “A novel hybrid feature selection method based on rough set and improved harmony search,” *Neural Comput. Appl.*, vol. 26, no. 8, pp. 1859–1880, 2015.

[28] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Micro Machine and Human Science, 1995. MHS’95., Proceedings of the 6th International Symposium*, 1995, pp. 39–43.

[29] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4.

[30] İ. B. Aydilek, “A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems,” *Appl. Soft Comput. J.*, vol. 66, pp. 232–249, 2018.

[31] K. Premalatha and a M. Natarajan, “Hybrid PSO and GA for Global Maximization,” *Int. J. Open Probl. Compt. Math*, vol. 2, no. 4, pp. 597–608, 2009.

[32] D. Chen, J. Chen, H. Jiang, F. Zou, and T. Liu, “An improved PSO algorithm based on particle exploration for function optimization and the modeling of chaotic systems,” *Soft Comput.*, vol. 19, no. 11, pp. 3071–3081, 2015.

[33] S. Q. Salih, A. A. Alsewari, B. Al-Khateeb, and M. F. Zolkipli, “Novel Multi-Swarm Approach for Balancing Exploration and Exploitation in Particle Swarm Optimization,” in *In Proceesdings of 3rd International Conference of Reliable Information and Communication Technology 2018 (IRICT 2018)*, 2018, pp. 196–206.

[34] S. Q. Salih and A. A. Alsewari, “Solving large-scale problems using multi-swarm particle swarm approach,” *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 1725–1729, 2018.

[35] S. Q. Salih, A. A. Alsewari, and Z. M. Yaseen, “Pressure Vessel Design Simulation: Implementing of Multi-Swarm Particle Swarm Optimization,” *Proc. 2019 8th Int. Conf. Softw. Comput. Appl.*, pp. 120–124, 2019.

[36] S. Q. Salih and A. A. Alsewari, “A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer,” *Neural Comput. Appl.*, vol. 32, no. 14, pp. 10359–10386, Jul. 2020.

[37] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” *Tech. Rep. TR06, Erciyes Univ.*, no. TR06, p. 10, 2005.

[38] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[39] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Micro Machine and Human Science, 1995. MHS’95., Proceedings of the Sixth International Symposium on*, 1995, pp. 39–43.

[40] UCI Machine Learning Repository, “Spambase Dataset,” *University of California, School of Information and Computer Science*. .

[41] S. Kumar, X. Gao, I. Welch, and M. Mansoori, “A Machine Learning based Web Spam Filtering Approach,” in *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*, 2016, pp. 973–980.

[42] K. R. D and P. Visalakshi, “A Hybrid ACO Based Feature Selection Method for Email Spam Classification,” *WSEAS TRANSACTIONS on COMPUTERS*, vol. 14. pp. 171–177, 2015.

[43] D. Karthika Renuka, P. Visalakshi, and T. Sankar, “Improving E-Mail Spam Classification using Ant Colony Optimization Algorithm,” *Int. J. Comput. Appl.*

[44] I. Idris, A. Selamat, N. Thanh 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.*, 2015.

[45] A. K. Uysal and S. Gunal, “A novel probabilistic feature selection method for text classification,” *Knowledge-Based Syst.*, 2012.