Feature Selection Approach based on Moth-Flame Optimization Algorithm

Hossam M. Zawbaa[1,2], E. Emary[3,4], B. PARV [1], Marwa Sharawi[4]
[1]Faculty of Mathematics and Computer Science, Babes-Bolyai University, Romania
[2]Faculty of Computers and Information, Beni-Suef University, Egypt
[3]Faculty of Computers and Information, Cairo University, Egypt
[4]Faculty of Computer Studies, Arab Open University, Cairo, Egypt

***Abstract*—In this work, a feature selection algorithm based on moth-flame optimization (MFO) is proposed. Moth-flame optimization (MFO) is a recently proposed swarm intelligent optimization algorithm that mimics the motion of moths. The proposed algorithm is applied in the domain of machine learning for feature selection to find the optimal feature combination using wrapper-based feature selection mode. In wrapper-based feature selection, a machine learning technique is used in the evaluation step. Despite it is very costly in time, this technique proved to have a good performance in classification accuracy. MFO is exploited in this study as a searching method to find optimal feature set, maximizing classification performance. The proposed algorithm is compared against particle swarm optimization (PSO) and genetic algorithms (GA). A set of UCI data sets is used for comparison using different assessment indicators. Results prove the efficiency of the proposed algorithm in comparison to other algorithms.**

Keywords*—*Bio-inspired Optimization, Moth-Flame Optimization, Feature Selection, Swarm Optimization.

I. Introduction

Feature selection is used for a better understanding of the generated data to select a subset of relevant features from a huge number of features. The search space size is exponentially increasing with respect to the number of features in the data set. Therefore, in practice the exhaustive search techniques are impossible to get the optimal solution and these feature selection techniques still suffer from stagnation in local optima [2]. The idea is that a selected feature set will improve the classifier performance and provide a faster classification, leading to comparable or even best classification accuracy than in the case we use all features [1].

There are two different approaches of feature selection that evaluates the quality of the selected features: wrapper-based and filter-based. More precisely, *wrapper-based approach* uses a machine learning technique to search through the space of possible solutions [1]. On the other hand, *filter-based approach* searches the feature space based on data-dependent criteria rather than classification-dependent criteria as in the wrapper approach [3].

A lot of heuristic algorithms mimic the behaviour of biological and physical systems in the nature and it has been proposed as strong methods for global optimizations. Evolutionary computation (EC) techniques are inspired from nature, social behavior, and biological behavior of animals or birds or insects like: bat, grey wolf, antlion, moth-flame, etc. in a group. Many researchers have proposed different computational methods, in order to mimic the behavior of these species to seek for their optimal solution [4]. Genetic algorithms (GA) and particle swarm optimization (PSO) are well-known evolutionary computation algorithms used in feature selection domain. GA is developed based on the natural process of evolution through reproduction [5]. In PSO, each solution is considered as a particle with specific characteristics (position, fitness, and speed vector) that defines the particle moving direction [6]. Artificial bee colony (ABC) is a numerical optimization algorithm based on the foraging behavior of honeybees. In ABC, the employer bees try to find a food source and the scout bees fly spontaneously to find the best food source [7]. The ant colony optimization (ACO) based wrapper feature selection algorithm was applied to network intrusion detection [8]. MFO algorithm is inspired from the moths navigation in nature, called transverse orientation [9].

The main objective of this paper is to propose a new MFO algorithm for feature selection problem that minimizes the number of selected features and maximizes the classification performance. The remainder of this paper is organized as the following: Section II provides the background information of MFO algorithm. Section III describes the proposed MFO algorithm for feature selection, while the experimental results with discussions are presented in section IV. Finally, conclusions and future work are provided in section V.

II. Preliminaries

A. Moth-Flame Optimization (MFO)

Moths have been evolved to fly in night using the moon light and they rely on a method called transverse orientation for navigation. In this method, a moth flies by maintaining a fixed angle with respect to the moon [9] (i.e. the light source). This method is considered a very effective technique for travelling long distances in a straight path [10]. Since the moon is far away from the moth, this method guarantees flying in straight line. When moths see a human-made artificial light, they try to maintain a similar angle with the light to fly in straight line. Since such a light is extremely close compared to the moon, however, maintaining a similar angle to the light source causes a useless or deadly spiral fly path for moths [11]. It may be observed that for artificial lights the moth eventually converges toward the light. An algorithm inspired from this type of motion is presented in the following section. Moths and flames are the main components of the algorithm. The candidate solutions are moths and the moths positions in space are the problem's variables. Therefore, moths can fly in 1-D, 2-D, 3-D, or even hyper-dimensional space (of dimension d) with changing position vectors. Since the MFO algorithm is a population-based algorithm, the set of n moths are used as search agents in the problem space. Flames are the best n positions of moths that are obtained so far. Therefore, each moth searches around a flag (flame) and updates it in case of finding a better solution. Therefore, flames are also d-dimensional data points. Given logarithmic spiral, a given moth updates its position with respect to a given flame as in equation (1).

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot cos(2\pi t) + F_j \tag{1}$$

where D_i means the Euclidian distance of the i^{th} moth for the j^{th} flame, b is a constant for defining the shape of the logarithmic spiral, M_i denotes the i^{th} moth, F_j is the j^{th} flame and t is a random number in $[-1, 1]$.

As may be seen in the above equation, the next position of a moth is defined with respect to a flame. In the spiral equation, t parameter defines how much the next position of the moth should be close to the flame. Therefore, a hyper-ellipse can be assumed around the flame in all directions and the next position of the moth would be within this space. In order to further emphasize exploitation, we assume that t is a random number in $[r, 1]$ where r is linearly decreasing from -1 to -2 over the course of iteration, being called *convergence constant*. With this method, moths tend to exploit their corresponding flames more accurately proportional to the number of iterations. In addition to enhance the probability of converging to a global solution, a given moth is obliged to update its position using only one of the flames. In each iteration and after updating the flames list, the flames are sorted based on their fitness values. The moths then update their positions with respect to their corresponding flames. To allow for much exploitation of the best promising solutions, the number of flames to be followed is decreased with respect to the iteration number as in equation (2). MFO is presented as in algorithm (1).

$$N_{flames} = round(N - l \cdot \frac{N-1}{T}) \tag{2}$$

where l is the current iteration number, N is the maximum number of flames, and T indicates the maximum number of iterations.

III. The proposed MFO feature selection algorithm

The proposed MFO for feature selection works in a wrapper-based manner. The main characteristic of wrapper methodologies is the use of the classifier as a guide of feature selection procedure. In this proposed algorithm, we used k-nearest neighbour (KNN) as a classification method to ensure the quality of the selected feature set [14]. KNN is a simple classification method that classifies the unknown instance based on the majority of the K-nearest neighbor category. Classifiers do not use any model for KNN and the classification is based on the minimum distance from the query unknown instance to the training samples [12].

input : n number of moths or search agents,
N maximum number of flames,
T number of iterations.
output: f_{best} optimal flame position,
$f(f_{best})$ fitness value for f_{best}.

1) Initialize a population of n flames positions randomly in the search space.
2) **while** *Stopping criteria not met* **do**
    Update the number of flames to be used N_{flames} according to equation 2.
    Calculate the fitness of all the n moths.
    if *first iteration* **then**
        Sort the moths from best to worst according to their fitness and place the result in flame matrix.
    else
        Merge the population of past moths and flames. Sort the merged population from best to worst. Select the best N positions from the sorted merged population as the flames.
    end
    Calculate the convergence constant r.
    foreach $Moth_i$ *with* $i \leq n$ **do**
        Calculate t as $t = (r-1) * rand + 1$. With $rand$ a random number drawn from uniform distribution in the range [0, 1].
        if $i \leq N$ **then**
            Update $Moth_i$ position according to $Flame_i$ using equation 1.
        else
            Update $Moth_i$ position according to $Flame_{N_{flames}}$ using equation 1.
        end
    end
end

Algorithm 1: Moth-flame optimization (MFO) algorithm

In this study, we made use of the recently proposed MFO algorithm to adaptively search the feature space for optimal feature combination, maximizing the classification performance. In MFO, moths continuously change their positions to whatever point in the space depending on the spiral equation 1. The t parameter in the equation is selected at random and decides the next position of a moth M_i. Spiral movement is the main component of the algorithm because it decides how the moths are repositioned around flames. The equation allows a moth to fly around a flame and not necessarily in the space between flames. For different values of t and given the distance between moth M_i and flame F_j, the new position of M_i can be seen as a change of the position of F_j and a change vector δ, which is given in equation (3).

$$\delta^t_{M_i, F_j, d} = D_i \cdot e^{bt} \cdot cos(2\pi t) \tag{3}$$

Individual solution is represented as a continuous vector with same dimension as number of features in the data set. The values in the solution vector are continuous and limited to the range [0, 1]. During the solution fitness evaluation, the continuous values of each solution are mapped to a binary (discrete) representation. The used objective function is

TABLE I. DATA SETS DESCRIPTION

Data set	No. of features	No. of samples
Breastcancer	9	699
Exactly	13	1000
Exactly2	13	1000
Lymphography	18	148
M-of-n	13	1000
Tic-tac-toe	9	958
Vote	16	300
Zoo	16	101
WineEW	13	178
SpectEW	22	267
SonarEW	60	208
PenglungEW	325	73
IonosphereEW	34	351
HeartEW	13	270
CongressEW	16	435
BreastEW	30	569
KrvskpEW	36	3196
WaveformEW	40	5000

TABLE II. PARAMETER SETTINGS

Parameter	Value
No. of search agents	8
No. of iterations	70
Problem dimension	Same as number of features
Search domain	[0 1]
b defining the shape of the logarithmic spiral in MFO	0.75
Crossover Fraction in GA	0.8
Inertia factor of PSO	0.1
Individual-best acceleration factor of PSO	0.1

usually reflecting the classification performance as well as the number of selected features. A generic representation of the fitness function representing both classification performance and number of selected features is given in equation (4).

$$f_\theta = \alpha \cdot E + (1-\alpha)\frac{\sum_i \theta_i}{N}, \tag{4}$$

where f_θ is the fitness function given a vector θ sized N with 0/1 elements representing unselected / selected features, N is the total number of features in the data set, E is the classifier error rate and α is a constant controlling the importance of classification performance to the number of features selected. In this study as the classification performance is the major goal we used α=1.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Data sets and Parameters

Table (I) summarizes the 18 used data sets for further experiments. The data sets are drawn from the UCI data repository [13]. The data is divided into 3 equal parts one for *training*, the second part is for *validation* and the third part is for *testing*. The training part is used to train the used classifier through optimization and at the final evaluation. The validation part assesses the performance of the classifier at the optimization time and the testing part is used to evaluate the finally selected features given the trained classifier. Three different optimization methods are compared in this study namely MFO, PSO, and GA to select appropriate feature subset and all algorithms are compared to the performance gained by using all features. The parameter setting for the three different optimization algorithms is outlined in table (II).

B. Evaluation Criteria

Each optimization algorithm is run for 20 times to test both (1) convergence capability of optimization algorithm and (2) the statistical significance, except for all features selected a solution that is forced to be a position for one of the search agents. Forcing the all features solution to guarantee that all subsequent feature subsets, if selected as a global best solution, are fitter than it. KNN is used in the experiments based on trial and error basis where the best choice of $(K = 5)$ is selected as the best performing on all the data sets [14]. The indicators used to compare the different algorithms are:

- *Mean test error:* is used to evaluate the performance of the feature selection on the data that the classifier never sees. The classification average error can be formulated in equation (5).

$$Test = \frac{1}{N}\sum_{j=1}^{N}\sqrt{\frac{1}{K}\sum_{i=1}^{K}(A_i - E_i)^2} \tag{5}$$

where K is the number of test sample points and A_i, E_i actual and expected class label for data point i.

- *Best fitness:* represents the minimum fitness function for each optimization algorithm at the different M operations of an optimization algorithm and can be formulated as in equation (6).

$$Best = min_{i=1}^{M} g_*^i, \tag{6}$$

where g_*^i is the optimal solution resulted from run number i.

- *Worst fitness:* represents the worst solution among the best solutions found for running each optimization algorithm for M times as in equation (7).

$$Worst = max_{i=1}^{M} g_*^i. \tag{7}$$

- *Mean fitness:* represents the average of solutions acquired from running an optimization algorithm for different M running as in equation (8).

$$Mean = \frac{1}{M}\sum_{i=1}^{M} g_*^i. \tag{8}$$

- *Standard deviation (std):* represents the variation of the best solutions found for running every optimization algorithm for M different runs as in equation (9).

$$Std = \sqrt{\frac{1}{M-1}\sum (g_*^i - Mean)^2}. \tag{9}$$

- *Average feature reduction:* represents the average size of the selected features to the total number of features and can be formulated as in equation (10).

$$Reduction = \frac{1}{M}\sum_{i=1}^{M}\frac{size(g_*^i)}{D}, \tag{10}$$

where $size(x)$ is the number of values for the vector x, D is the number of features in the original data set, and g_*^i is the optimal solution resulted from run number i.

- *Average fisher score (f-score):* evaluates the data space spanned by the selected features, which means the distances between data points in different classes are as large as possible, while the distances between data points in the same class are as small as possible, as shown in equation (11) [15].

$$F_j = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \mu^j)^2}{(\sigma^j)^2}, \tag{11}$$

where F_j is the fisher score for feature j, $\mu^j, (\sigma^j)^2$ is the mean and std of the whole data set, n_k is the size of class k, and μ_k^j is the mean of class k.

C. Results and discussion

Table (III) outlines the average statistical mean for each optimization algorithms over the different data sets calculated over the 20 runs. We can remark that all used optimization algorithms outperform all the features, proving the capability of the wrapper-based method in feature selection problem. We can also highlight that the MFO performs, in general, better than GA and PSO, proving the capability of MFO to adaptively search the feature space for optimal feature combination and its ability to avoid premature convergence that may be caused by falling in local minima. Same conclusion can be confirmed by remarking the performance on tables (IV), (V), and (VI).

Table (VII) outlines the performance of the selected features from the optimization algorithms in the classification of test data averaged over the 20 runs. We can see from the table that 11 data sets achieve its best performance with the selection of MFO which proves the capability of MFO to find optimal feature combination ensuring good testing performance. Regarding the size of selected features with respect to the original size, table (VIII) outlines the kept feature ratio to the total number of features. We can highlight that MFO outperforms the other methods in classification performance, but also has a comparable ratio of features selected, which confirms that the MFO can select the optimal feature combination with comparable size. The performance over the test data is to some extent compatible with the results obtained from the fisher score calculated over the selected features by the different optimizers; as shown in the table (IX).

V. Conclusion and future work

The aim of this paper was to propose a moth-flame optimization (MFO) for feature selection to select the minimal number of features and obtaining comparable or even better classification accuracy from utilizing all features. We applied MFO in a wrapper-based manner for feature selection using classification performance as fitness function. The proposed algorithm is compared to particle swarm optimization (PSO)

TABLE III. Mean fitness for the different optimization algorithms

Data set	All	GA	PSO	MFO
Breastcancer	0.040	**0.026**	0.030	0.028
BreastEW	0.061	0.033	0.036	**0.032**
CongressEW	0.074	**0.033**	0.036	0.044
Exactly	0.332	0.283	0.286	**0.243**
Exactly2	0.249	0.258	0.251	**0.237**
HeartEW	0.176	0.147	0.171	**0.118**
IonosphereEW	0.157	0.147	0.138	**0.115**
KrvskpEW	0.087	0.047	**0.047**	0.049
Lymphography	0.269	**0.152**	0.180	0.163
M-of-n	0.169	0.108	**0.073**	0.096
PenglungEW	0.285	**0.206**	0.244	0.227
SonarEW	0.333	0.157	0.169	**0.142**
SpectEW	0.220	**0.135**	0.144	0.151
Tic-tac-toe	0.264	0.235	**0.231**	0.240
Vote	0.094	**0.040**	0.040	0.044
WaveformEW	0.238	0.218	0.216	**0.206**
WineEW	0.061	0.020	0.023	**0.014**
Zoo	0.219	0.103	**0.099**	0.100
Total	3.330	2.348	2.414	**2.248**

TABLE IV. Best fitness for the different optimization algorithms

Data set	All	GA	PSO	MFO
Breastcancer	0.030	0.017	0.022	**0.013**
BreastEW	0.058	**0.021**	0.026	0.021
CongressEW	0.048	0.021	**0.014**	0.028
Exactly	0.308	0.258	0.189	**0.087**
Exactly2	0.228	0.234	0.231	**0.216**
HeartEW	0.144	**0.089**	0.133	0.111
IonosphereEW	0.137	0.120	0.111	**0.103**
KrvskpEW	0.067	0.032	0.036	**0.032**
Lymphography	0.204	0.140	0.140	**0.082**
M-of-n	0.132	0.084	0.018	**0.009**
PenglungEW	**0.042**	0.120	0.160	0.042
SonarEW	0.261	0.100	**0.086**	0.101
SpectEW	0.180	**0.079**	0.101	0.112
Tic-tac-toe	0.231	**0.204**	**0.204**	0.213
Vote	0.060	0.030	**0.010**	0.040
WaveformEW	0.230	0.202	0.202	**0.189**
WineEW	0.034	0.017	0.017	**0.000**
Zoo	0.125	0.063	0.069	**0.030**
Total	2.519	1.830	1.770	**1.428**

TABLE V. Worst fitness for the different optimization algorithms

Data set	All	GA	PSO	MFO
Breastcancer	0.060	**0.030**	0.039	0.039
BreastEW	0.068	0.047	0.058	**0.042**
CongressEW	0.097	**0.048**	0.062	0.062
Exactly	0.362	0.312	0.342	**0.299**
Exactly2	0.266	0.270	0.270	**0.254**
HeartEW	0.200	0.178	0.189	**0.122**
IonosphereEW	0.188	0.171	0.162	**0.137**
KrvskpEW	0.110	**0.061**	0.065	0.068
Lymphography	0.367	**0.180**	0.220	0.306
M-of-n	0.192	0.156	**0.117**	0.147
PenglungEW	0.417	0.320	**0.320**	0.417
SonarEW	0.406	0.214	0.229	**0.174**
SpectEW	0.270	**0.180**	0.191	0.180
Tic-tac-toe	0.300	0.260	0.260	**0.250**
Vote	0.130	**0.050**	0.080	0.060
WaveformEW	0.249	0.244	0.224	**0.218**
WineEW	0.085	**0.033**	0.033	0.034
Zoo	0.353	**0.152**	0.152	0.235
Total	4.119	**2.908**	3.014	3.044

TABLE VI. STANDARD DEVIATION FOR THE DIFFERENT OPTIMIZATION ALGORITHMS

Data set	All	GA	PSO	MFO
Breastcancer	0.013	**0.005**	0.007	0.009
BreastEW	**0.005**	0.010	0.013	0.008
CongressEW	0.021	**0.014**	0.020	0.017
Exactly	0.021	**0.021**	0.058	0.090
Exactly2	0.018	**0.015**	0.018	0.015
HeartEW	0.021	0.040	0.022	**0.006**
IonosphereEW	0.020	0.020	0.023	**0.014**
KrvskpEW	0.016	0.012	**0.011**	0.014
Lymphography	0.074	**0.018**	0.028	0.089
M-of-n	**0.027**	0.029	0.046	0.054
PenglungEW	0.159	0.088	**0.075**	0.144
SonarEW	0.063	0.057	0.060	**0.026**
SpectEW	0.041	0.038	0.036	**0.031**
Tic-tac-toe	0.032	0.022	0.022	**0.016**
Vote	0.029	0.010	0.027	**0.009**
WaveformEW	**0.007**	0.016	0.008	0.011
WineEW	0.019	**0.007**	0.009	0.014
Zoo	0.085	0.034	**0.031**	0.082
Total	0.672	**0.456**	0.515	0.652

TABLE VII. MEAN TEST ERROR FOR THE DIFFERENT OPTIMIZATION ALGORITHMS

Data set	All	GA	PSO	MFO
Breastcancer	0.039	0.026	0.030	**0.025**
BreastEW	0.055	**0.033**	0.036	**0.033**
CongressEW	0.079	**0.033**	0.036	0.037
Exactly	0.338	0.283	0.286	**0.278**
Exactly2	0.262	0.258	**0.251**	0.257
HeartEW	0.218	**0.147**	0.171	0.149
IonosphereEW	0.188	0.147	0.138	**0.137**
KrvskpEW	0.093	0.047	0.047	**0.046**
Lymphography	0.288	0.152	0.180	**0.136**
M-of-n	0.144	0.108	**0.073**	0.099
PenglungEW	0.285	**0.206**	0.244	0.214
SonarEW	0.329	0.157	0.169	**0.131**
SpectEW	0.204	0.135	0.144	**0.130**
Tic-tac-toe	0.276	0.235	**0.231**	0.234
Vote	0.092	0.040	0.040	**0.034**
WaveformEW	0.241	0.218	0.216	**0.210**
WineEW	0.060	**0.020**	0.023	0.023
Zoo	0.189	0.103	0.099	**0.091**
Total	3.381	2.348	2.414	**2.265**

TABLE VIII. AVERAGE SELECTION SIZE FOR THE DIFFERENT OPTIMIZATION ALGORITHMS

Data set	GA	PSO	MFO
Breastcancer	**0.511**	**0.511**	0.578
BreastEW	0.460	**0.440**	0.487
CongressEW	**0.362**	0.400	0.388
Exactly	0.523	**0.492**	0.600
Exactly2	**0.215**	0.462	0.338
HeartEW	**0.508**	0.538	0.569
IonosphereEW	**0.418**	0.424	0.500
KrvskpEW	0.539	0.533	**0.511**
Lymphography	**0.400**	**0.400**	0.411
M-of-n	0.677	0.600	**0.538**
PenglungEW	0.434	**0.387**	0.436
SonarEW	**0.420**	0.450	0.463
SpectEW	**0.373**	0.464	0.445
Tic-tac-toe	**0.556**	**0.556**	0.622
Vote	**0.362**	0.512	0.400
WaveformEW	0.580	**0.520**	0.550
WineEW	0.477	0.523	**0.415**
Zoo	0.362	0.400	**0.325**
Total	**8.178**	8.612	8.578

TABLE IX. AVERAGE FISHER SCORE FOR THE DIFFERENT OPTIMIZATION ALGORITHMS

Data set	GA	PSO	MFO
Breastcancer	0.713	**0.692**	0.829
BreastEW	0.245	0.230	**0.219**
CongressEW	**0.201**	0.219	0.219
Exactly	0.001	**0.001**	0.001
Exactly2	**0.001**	0.001	0.001
HeartEW	**0.074**	0.079	0.101
IonosphereEW	0.033	**0.032**	0.040
KrvskpEW	0.022	0.021	**0.021**
Lymphography	**0.167**	0.177	0.170
M-of-n	0.030	0.031	**0.027**
PenglungEW	0.330	**0.285**	0.330
SonarEW	0.019	**0.018**	0.021
SpectEW	**0.021**	0.027	0.025
Tic-tac-toe	**0.005**	**0.005**	0.005
Vote	**0.184**	0.214	0.191
WaveformEW	0.131	**0.121**	0.131
WineEW	0.504	0.511	**0.450**
Zoo	12.298	13.406	**9.609**
Total	14.979	16.071	**12.391**

and genetic algorithms (GA) using different evaluation criteria on 18 different data sets from UCI machine learning repository. The experiment results proves the capability of MFO to adaptively search the feature space to find optimal feature combination maximizing classification accuracy. In addition to, the results proves that the performance of MFO is significantly better than GA and PSO which are common in wrapper-based feature selection. On the basis of future performance, the proposed method will be assessed using more complex data sets and we will apply different evaluation measures such as computational time.

ACKNOWLEDGMENT

This work was partially supported by the IPROCOM Marie Curie initial training network, funded through the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement No. 316555.

REFERENCES

[1] I. Guyon, A. Elisseeff, "An introduction to variable and attribute selection", Machine Learning Research, Vol. 3, pp. 1157-1182, 2003.

[2] B. Xue, M. Zhang, W.N. Browne. "Particle swarm optimization for feature selection in classification: a multi-objective approach", IEEE transactions on cybernetics, Vol. 43, No. 6, pp. 1656-1671, 2013.

[3] B. Xue, M. Zhang, W.N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms", Applied Soft Computing, Vol. 18, pp. 261-276, 2014.

[4] S. Shoghian, M. Kouzehgar, "A Comparison among Wolf Pack Search and Four other Optimization Algorithms", Computer, Electrical, Automation, Control and Information Engineering, Vol. 6, No. 12, pp. 1619-1624, 2012.

[5] C.L. Huang, C.J. Wang, "A GA-based feature selection and parameters optimization for support vector machines", Expert Systems with Applications, Vol. 31, No. 2, pp. 231-240, 2006.

[6] C.L. Huang, J.F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization", Applied Soft Computing, Vol. 8, No. 4, pp. 1381-1391, 2008.

[7] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Global Optimization, Vol. 39, No. 3, pp. 459-471, 2007.

[8] H. Ming, "A rough set approach to feature selection based on ant colony optimization", Pattern Recognition Letter, 2010.

[9] S. Mirjalili, ”Moth-Flame Optimization Algorithm: A Novel Nature-inspired Heuristic Paradigm”, Knowledge-Based Systems, Vol. 89, pp. 228-249, 2015.

[10] K.J. Gaston, J. Bennie, T.W. Davies, J. Hopkins, ”The ecological impacts of nighttime light pollution: a mechanistic appraisal”, Biological reviews, Hindawi Publishing Corporation, Vol. 88, No. 1, pp. 912-927, 2013.

[11] K.D. Frank, C. Rich, T. Longcore, ”Effects of artificial night lighting on moths”, Journal of Ecological consequences of artificial night lighting, Hindawi Publishing Corporation, Vol. 2015, No. 1, pp. 305-344, 2006.

[12] A.S. Anwar, K.K.A. Ghany, H. Elmahdy, ”Human Ear Recognition Using Geometrical Features Extraction”, International Conference on Communication, Management and Information Technology, Procedia Computer Science, Vol. 65, pp. 529-537, 2015.

[13] A. Frank, A. Asuncion, ”UCI Machine Learning Repository”, 2010.

[14] L.Y. Chuang, H.W. Chang, C.J. Tu, C.H. Yang, ”Improved binary PSO for feature selection using gene expression data”, Computational Biology and Chemistry, Vol. 32, pp. 29-38, 2008.

[15] R.O. Duda, P.E. Hart, D.G. Stork, ”Pattern Classification”, 2nd edition, Wiley-Interscience Publication, 2000.