

# MODIFIED MOTH-FLAME OPTIMIZATION ALGORITHMS FOR TERRORISM PREDICTION

Ghada M. A. Soliman<sup>1</sup>, Motaz M. H. Khorshid<sup>2</sup>, Tarek H. M. Abou-El-Enien<sup>3</sup>

<sup>1,2,3</sup> DS & OR department, Faculty of Computers and Information, Cairo University, Egypt

## ABSTRACT

*In this work, new versions of moth-flame optimization algorithm are proposed and used to select optimal feature subset for the classification purposes. Moth-flame Optimization (MFO) algorithm is one of the newest bio inspired optimization techniques in which the main inspiration of this optimizer is the navigation method of moths in nature called transverse orientation. Moths fly in night by maintaining a fixed angle with respect to the moon, a very effective mechanism for travelling in a straight line for long distances. However, these fancy insects are trapped in a useless/deadly spiral path around artificial lights. This paper presents two modified versions of MFO algorithm which used as a prediction method for terrorist groups. The two modified versions of MFO are compared with the original MFO as well as compared to other well known nature-inspired algorithms as Ant Lion Optimizer (ALO) algorithm, Grey Wolf Optimization (GWO) algorithm, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Two different experiments modes are conducted and two well-known classification algorithms are used in the classification process; Random Forests (RF) ensemble classifier and K-Nearest Neighbor (KNN) algorithm. A set of assessment indicators are used to evaluate and compare between the obtained results which prove that the proposed modified versions of MFO provide very promising and competitive performance as well as achieve an advance over the original MFO algorithm with high stability over other searching methods.*

**Keywords:** Moth-Flame Algorithm, Random-Forests, Terrorism Prediction

## 1. INTRODUCTION

Over the last few decades; the optimization refers to the process of finding the best possible solution(s) for a particular problem. As the complexity of problems real applications increases, the need for new optimization techniques becomes evident more than before. Mathematical optimization techniques used to be the only tools for optimizing problems before the proposal and existence of heuristic optimization techniques.

Mathematical optimization methods are mostly deterministic that suffer from a major problem which is local optima entrapment [1]. Some of them such as gradient-based algorithms require derivation of the search space as well. This makes them highly inefficient in solving real problems. Due to this limitation; some of heuristic algorithms begin to be proposed, started from Genetic Algorithm (GA) [2] which is the most popular stochastic optimization algorithm and due to its mechanism in solving the problem depends on its stochastic components in the selection, re-production, and mutation phases so this help GA to avoid local optima, another key success of GA is the overall average fitness of population which is improved over the course of generations, these points This makes GA highly suitable for solving real problems with unknown search spaces. Researchers gave high attention to such evolutionary search algorithms after GA, that result in proposing different and multiple well known algorithms like Particle Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5], Evolutionary Strategy (ES) [6], and Evolutionary Programming (EP) [7]-[8]. The application of these algorithms can be found in different branches of science and industry real applications. According to the No-Free-Lunch (NFL) theorem [9], there is no algorithm for solving all optimization problems. This means that an optimizer may perform well in a set of problems and fail to solve a different set of problems. In other words, the average performance of optimizes is equal when considering all optimization problems. Therefore, there are still problems that can be solved by new optimizers better than the current optimizers. Recently, there are new inspiration algorithms that have been proposed, one of the most new algorithms that developed by Seyedali Mirjalili in [1] is Moth-Flame Optimization (MFO) algorithm. The main inspiration of this optimizer is the navigation method of moths in nature called transverse orientation. Moths fly in night by maintaining a fixed angle with respect to the moon, a very effective mechanism for travelling in a straight line for long distances. However, these fancy insects are trapped in a useless/deadly spiral path around artificial lights.

Our research study first proposes the mathematical modification for MFO algorithm into two novel (modified) versions of spiral flying path of moths around artificial lights (flames), and then we used the new versions of MFO in a prediction algorithm to predict the responsible terrorist groups responsible for the terrorist attack in Egypt.

The two modified versions of MFO are compared with the original MFO as well as compared to other well known nature-inspired algorithms as Ant Lion Optimizer (ALO) algorithm, Grey Wolf Optimization (GWO) Algorithm, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Two different experiments modes are conducted and two well-known classification algorithms are used in the classification process; Random Forests (RF) ensemble classifier and K-Nearest Neighbor (KNN) algorithm. A set of assessment indicators are used to evaluate and compare between the obtained results which prove that the proposed modified versions of MFO provide very promising and competitive performance as well as achieve an advance over the original MFO algorithm with high stability over other searching methods. The remainder of this paper is organized as follows. Section 2 provides illustration of a moth-flame optimization (MFO) algorithm. Section 3 describes in details the new versions of MFO algorithm (MFO2, MFO3). Section 4 provides the full details of the different phases of the proposed hybrid MFO2-RFs classification prediction algorithm. Section 5 presents the experimental results and analysis of the proposed system. Section 6 provides Conclusions and future work.

## 2.MOTH-FLAME OPTIMIZATION (MFO) ALGORITHM

In this algorithm [10], it is assumed that the candidate solutions are moths and the problem's variables are the position of moths in the space. Therefore, the moths can fly in 1-D, 2-D, and 3-D with changing their position vectors. Since the MFO algorithm is a population-based algorithm. The moths are actual search agents that move around the search space, whereas flames are the best position of moths that obtains so far. In other words, flames considered as flags or pins that are dropped by moths when searching the search space. Therefore, each moth searches around a flag (flame) and updates it in case of finding a better solution. The algorithm author chose the logarithmic spiral as the main update mechanism of moths. However, any types of spiral can be utilized, where subject to the following conditions:

1. Spiral's initial point should start from the moth
2. Spiral's final point should be the position of the flame
3. Fluctuation of the range of spiral should not exceed from the search space

### 2.1 Mathematical Formulation

The main key components are moths and flames which illustrated as follows:

#### 2.1.1 Moths key Components

1. The MFO algorithm is a three-tuple that approximate the global optimal of the optimization problems and defined as follows:

$$\text{MFO} = (\text{I}, \text{P}, \text{T}) \quad (1)$$

Where:

- **I**: A function that generates a random population of moths around the search space, this function received the matrix of  $M$  and returns its updated one eventually.

$$\text{I}: \emptyset \rightarrow \{\text{M}, \text{OM}\} \quad (2)$$

- $M$ : is a matrix used to represent the months with their position

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,d} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,d} \end{bmatrix} \quad (3)$$

$n$ : The number of moths

$d$ :The number of variables (dimension)

- $OM$ :the array for sorting the corresponding fitness values for all the moths as follows:

$$OM = \begin{bmatrix} OM_1 \\ \vdots \\ OM_n \end{bmatrix} \quad (4)$$

$n$ : The number of moths

The fitness value is the return value of the fitness (objective) function for each moth. The position vector of each moth in  $M$  matrix is passed to the fitness function and the output of the fitness function is assigned to the corresponding moth as its fitness value in  $OM$  matrix.

- $P$ : Function is the main function; moves the moths around the search space, this function received the matrix  $M$  and returns its updated one eventually

$$P: M \rightarrow M \quad (5)$$

- $T$ : function returns true if the termination criterion is satisfied and false if the termination criterion is not met

$$T: M \rightarrow \{\text{true, false}\} \quad (6)$$

2. The General framework of the MFO is defined as follows:

```

M = I();
While T(M) is equal to false
    M = P (M)
end
    
```

### 2.1.2 Flames key Components

- $F$ : is a matrix used to represent their flames with their positions that include  $n$  recent best solutions obtained so far in order to improve the probability of finding better solutions.

$$F = \begin{bmatrix} F_{1,1} & \cdots & F_{1,d} \\ \vdots & \ddots & \vdots \\ F_{n,1} & \cdots & F_{n,d} \end{bmatrix} \quad (7)$$

$n$ : The number of flames

$d$ : The number of variables (dimension)

- $OF$ : the array for sorting the corresponding fitness values for the flames

$$OF = \begin{bmatrix} OF_1 \\ \vdots \\ OF_n \end{bmatrix} \quad (8)$$

$n$ : The number of moths.

### 2.1.3 Generation of Initial Solutions

The function  $I$  is corresponding for generation of initial solutions and calculate the objective values where any random distribution can be used as presented below:

```

for i = 1:n
    for j = 1:d
        M(i,j) = (ub(i) - lb(i)) * rand() + lb(i);
    end
end
OM = FitnessFunction(M);
    
```

- $ub$ : is the upper bound of the variables

$$ub = [ub_1, ub_2, \dots, ub_{n-1}, ub_n] \quad (9)$$

Where  $ub_i$  indicates the upper bound of the  $i$ -th variable.

- $lb$ : is the lower bound of the variables

$$lb = [lb_1, lb_2, \dots, lb_{n-1}, lb_n] \quad (10)$$

Where  $ub_i$  indicates the upper bound of the  $i$ -th variable.

**2.1.4. Transverse Orientation Behaviour of Moths**

As the inspiration of MFO algorithm is the transverse orientation of moths, so the position of each moth is updated with respect to a flame using Eqn. (11) as follows:

$$M_i = S(M_i, F_j) \quad (11)$$

Where  $M_i$  indicates the  $i$ -th moth,  $F_j$  indicates the  $j$ -th flame, and  $S$  is the spiral function (movement), which is the main component of the algorithm because it dictates how the moths update their position around flames. The spiral equation allows a moth to fly “around” a flame and not necessarily in the space between them.

A logarithmic spiral is chosen for the original MFO algorithm, which is defined as follows:

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad (12)$$

Where  $D_i$ : The distance of the  $i$ -th moth for the  $j$ -th flame.

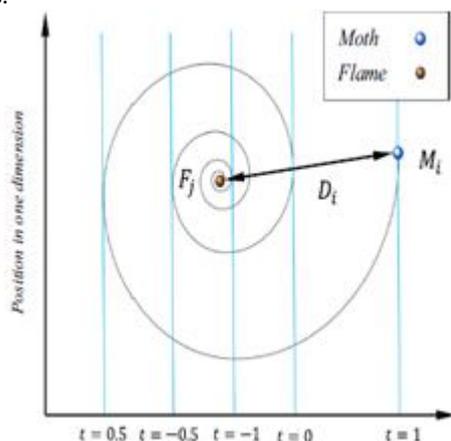
$b$ : A constant for defining the shape of the logarithmic spiral.

$t$ : A random number in  $[-1, 1]$ , and defines how much the next position of moth should be close to the flame. ( $t = -1$ ) is the closest position to the flame, while ( $t = 1$ ) shows the farthest.

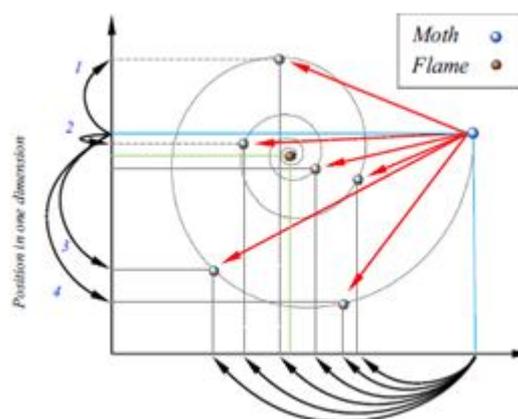
$D$  is calculates as follows:

$$D_i = |F_j - M_i| \quad (13)$$

By this method the moths tend to exploit their corresponding flames more accurately proportional to the number of iterations.



**Figure 1** Logarithmic spiral, space around a flame, and the position with respect to .



**Figure 2** some possible positions that can be reached by a moth w.r.t. a flame using a logarithmic spiral

**3. The General Steps of P function**

```

update flame No. using Eqn. (14)
OM = FitnessFunction (M) ;
if iteration == 1
    F = sort (M);
    OF = sort (OM);
else
    F = sort (M_{t-1}, M_t );
    OF = sort (M_{t-1}, M_t);
end
for i = 1: n
    for j = 1: d
        Update r and t
        Calculate D using Eqn. (13) with respect to the corresponding moth.
        Update M(i, j) using Eqns. (11) and (12) with respect to the corres. moth
    end
end
end
    
```

**2.2 Some Interesting Points about MFO Algorithm**

- A moth can converge to any point in the neighbourhood of the flame by changing
- The lower values of  $t$ , the closer distance to the flame.
- The frequency of position updating on both sides of the flame is increased as the moth get closer to the flame.
- In order to further emphasize exploitation; it is assumed that  $r$  is a random number in  $[r, 1]$  where:  $r$  is the convergence constant and it is decreased from -2 to -1 over the course of iteration.
- According to the values of the parameter  $r$ , a hyper ellipse can be assumed around the flame in all directions and the next position of the moth would be within this space.
- Each moth is obligated to update its position using only one of the flames in Eqn. 12, in order to prevent MFO algorithm to be trapped in a local optima quickly. The reason that a specific flame is assigned to each moth is to prevent local optimum stagnation.
- Moths update their position in hyper spheres around the best solutions obtained so far.
- The sequence of flames if changed based on the best solutions in each iteration, and the moths are required to update their positions with respect to the updated flames. Therefore, the position updating of moths may occur around different flames, a mechanism that causes sudden movement of moths in the search space and promotes exploration.
- The position updating of moths with respect to  $n$  different locations in the search space may degrade the exploitation of the best promising solutions. To resolve this concern and adaptive mechanism is proposed for the number of flames (which is decreased iteratively over the course of iteration).

$$flame\ Number = round\left(N - 1 * \frac{N-1}{T}\right) \tag{14}$$

**3. NEW VERSIONS OF MFO ALGORITHM (MFO2, MFO3)**

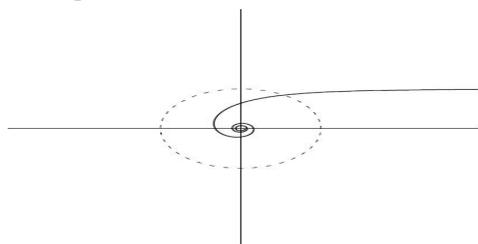
As mentioned in the original MFO algorithm in [10] that transverse orientation behaviour of moths is the main component that simulates the movement of moths in the hyper sphere space around the flames which is explained by (11). A logarithmic spiral is chosen for the original MFO algorithm as a spiral movement of moths around the flames which is defined in (12). According to that reason; in our research work we intended to try different spirals and investigate the performance of the updating in the original MFO algorithm by conducting different experiments in a hybrid terrorism prediction system to predict the terrorist group(s) responsible of attacks in Egypt during the period from year 2006 to 2014. The results of MFO2, and MFO3 are tested and compared with the original MFO algorithm as well as to other well-known search algorithms which proved high stability and comparative performance in the search space such as Grey wolf optimization (GWO) algorithm, Ant lion Optimizer (ALO), Genetic Algorithms (GA) especially GA with multi-parent recombination, and Particle Swarm Optimization (PSO) algorithm.

**3.1 Moth-Flame Optimization-2 (MFO2) Algorithm**

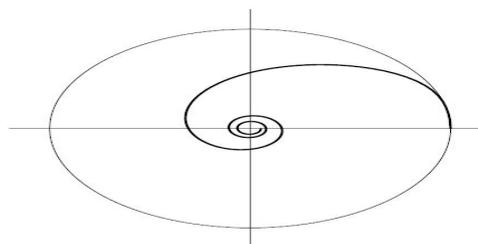
In this new version of the MFO algorithm, we based on using the hyperbolic (reciprocal) spiral. A hyperbolic spiral is a transcendental plane curve also known as a reciprocal spiral [10]. A hyperbolic spiral is the opposite of an Archimedean spiral with inversion center at the origin, and is a type of Cotes' spiral.

**3.1.1 Spiral Definition**

The hyperbolic (reciprocal) spiral begins at an infinite distance from the pole in the center (for  $\theta$  starting from zero  $r = a/\theta$  starts from infinity), and it winds faster and faster around as it approaches the pole; the distance from any point to the pole, following the curve, is infinite as shown in figure 3 and figure 4 that represent reciprocal spiral and its hyperbolic counterpart.



**Figure 3** The reciprocal spiral  $r = a/\theta$



**Figure 4** The hyperbolic version of the Reciprocal spiral

**3.1.2 Mathematical Formulation**

In the original MFO algorithm we can substitute the logarithmic Spiral function which as explained in Eqn. (15) with our reciprocal (hyperbolic) function and update it to reflect the moth movement around the flame with a specific direction in MFO2 algorithm as shown in Eqn. (15). As follows:

$$S(M_i, F_j) = D_i \cdot \frac{\cos(2\pi t)}{t} + F_j \quad (15)$$

Where  $S$ : is the Hyperbolic spiral function (movement), which is the main component of the algorithm.

$M_i$  : Indicates the i-th moth as explained in Eqn.(3),

$F_j$  : Indicates the j-th flame as explained in Eqn.(7),

$D_i$ : The distance of the i-th moth for the j-th flame, and calculated in Eqn. (13)

$t$ : A random number in  $[-1, 1]$ , and defines how much the next position of moth should be close to the flame. ( $t = -1$ ) is the closest position to the flame, while ( $t = 1$ ) shows the farthest.

**3.2 Moth-Flame Optimization-3 (MFO3) Algorithm**

In our research study we apply a different spiral function called Archimedes' Spiral in which it is a member of the big spiral family called Archimedean (Arithmetic) spiral.

**3.2.1. Spiral Definition**

The Archimedean spiral is a curve starts in the origin and makes a curve with three rounds where the distance between the spiral branches are the same. More exact: The distances of intersection points along a line through the origin are the same as in figure 5, and defined by a polar equation of the form:

$$r = a\theta^{1/n} \quad (16)$$

Where for different values of  $n$  we get different spirals ( $n$  is a constant that determines how tightly the spiral is wrapped). For  $n=1$ , we get the Archimedes' Spiral which explained in Figure 6.

The Archimedes' Spiral has a polar equation form:

$$r = a\theta \quad (17)$$

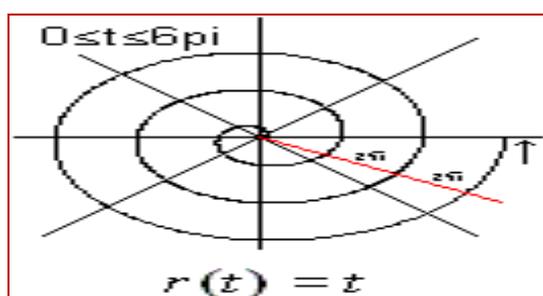


Figure 5 The Archimedean. Spiral

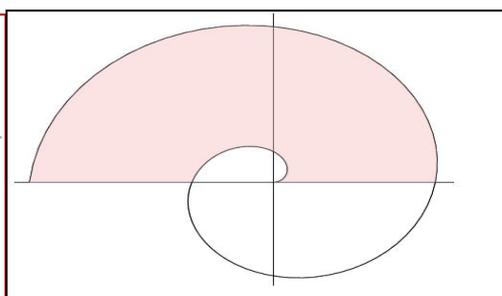


Figure 6 the Archimedes' Spiral.

**3.2.2 Mathematical Formulation**

In the original MFO algorithm we substituted the logarithmic Spiral function which as explained in Eqn. (15) with our Archimedes' Spiral function and update it to reflect the moth movement around the flame with a specific direction in MFO3 algorithm as shown in Eqn. (18). As follows:

$$S(M_i, F_j) = D_i \cdot t \cos(2\pi t) + F_j \quad (18)$$

Where  $S$ : is the Archimedes' spiral function (movement), which is the main component of the algorithm.

$M_i$ : Indicates the i-th moth as explained in Eqn.(3),

$F_i$  : Indicates the j-th flame as explained in Eqn.(7),

$D_i$ : The distance of the i-th moth for the j-th flame, and calculated in Eqn. (13)

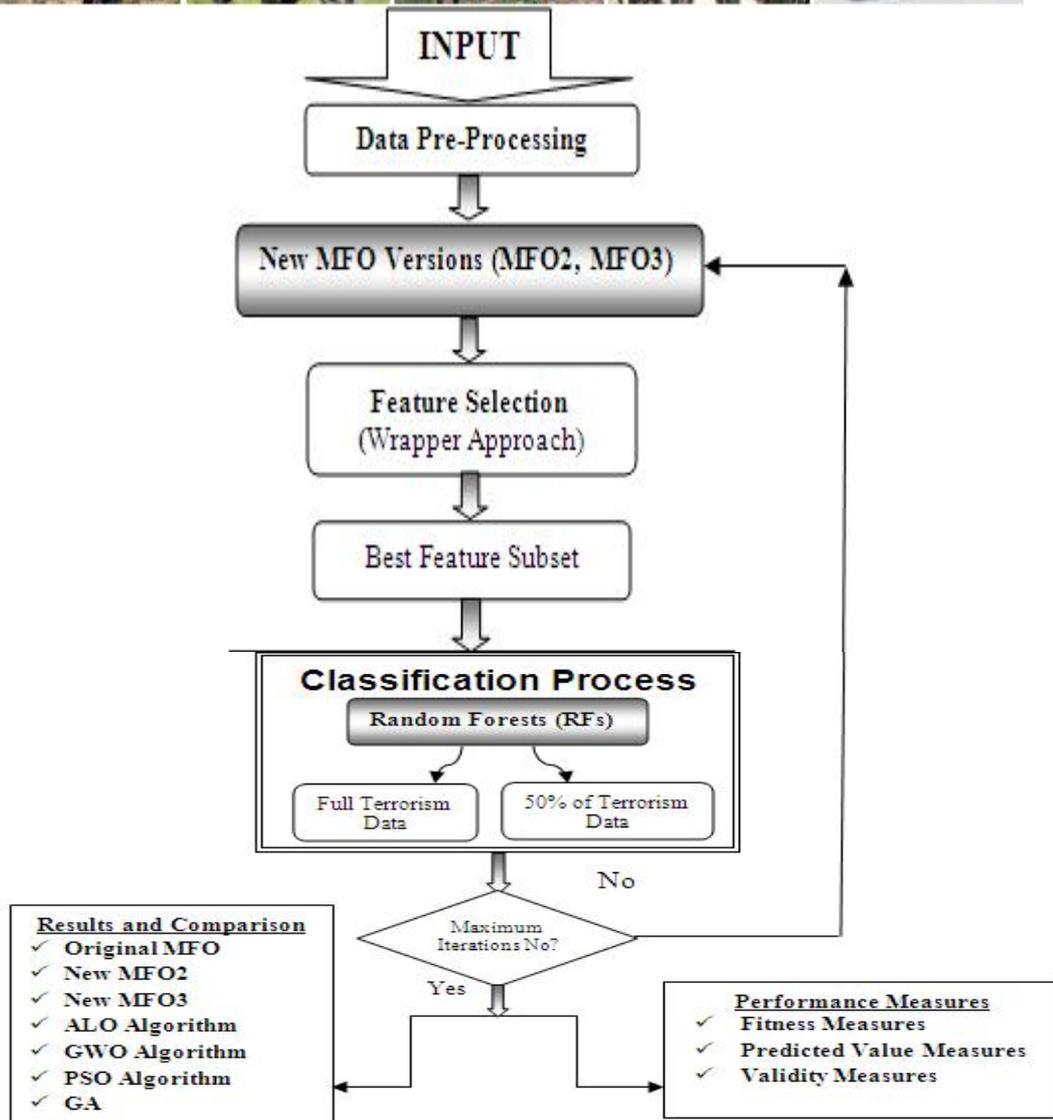
$t$ : A random number in  $[-1, 1]$  and defines how much the next position of moth should be close to the flame.(  $t = -1$ ) is the closest position to the flame, while ( $t = -1$ ) shows the farthest.

**4. PROPOSED NEW HYBRID PREDICTION ALGORITHM**

The proposed ensemble classification prediction algorithm consists of different phases as explained in Figure7.

**Algorithm 1: The Proposed Ensemble Classification Algorithm:**

1. Terrorism Data Pre-Processing,
2. Apply the MFO2 (or MFO3) ,
3. Feature Selection (Apply the Wrapper Approach),
4. Classification Process (Apply the Random Forests ensemble classification Algorithm) as well as apply K-Nearest Neighbor for the comparison purpose,
5. Stopping Criterion (If No. of iterations > *Max iteration No.*, then Stop, and go to Step 6; otherwise go to step2).
6. Results and Comparison,
7. Evaluate Performance Measures.



**Figure 7** Proposed New Hybrid Prediction Algorithm Framework

## 5. EXPERIMENTAL RESULTS and ANALYSIS

In our research study we conducted two different experiments using the data about terrorist attacks on Egypt during the period from 2006 to 2014. The data we based on are real world data from Global Terrorism Database (GTD).

We conduct the first experiment based on using the full dataset while the second one based on using 50% of the full data. In this paper, a set of assessment measures are used to compare between the original MFO and other two new versions (MFO2, MFO3) along with applying two different classification algorithms (RFs, and KNN).

### 5.1 Results and Different Comparisons.

It is obvious from the results of TABLE1 that the fitness performance in first experiment of MFO2 is the best corresponding to original MFO, and MFO3 as also explained in Figure 8; by comparing the results with other optimizers we can notice that MFO2 gives very promising and comparative results with respect to ALO, GWO, and GA.

According to the results from the second experiment that conducted by using 50% of the terrorism data; we can observe from Figure 9 that MFO2 still gives remarkable results over MFO, and performs very competitively with respect to GWO and ALO, that is confirm the searching capability of MFO2.

**TABLE 1:** The Fitness Results of the experiments by using KNN Classifier

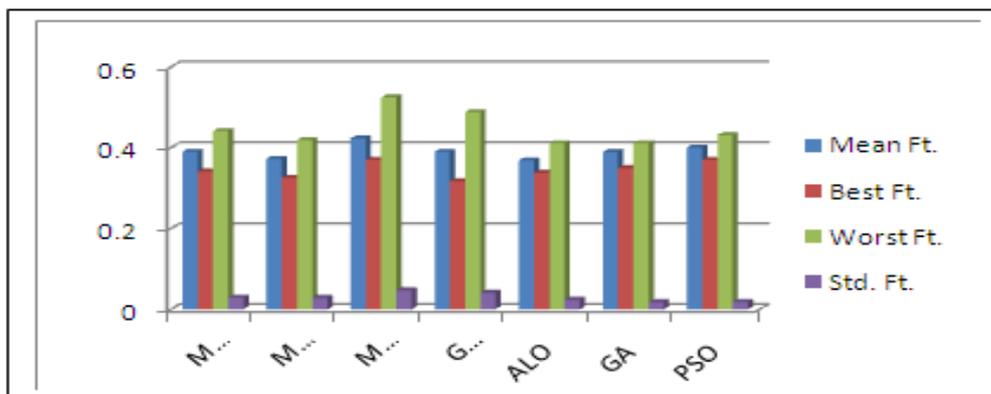
Experiment	Measure	MFO	MFO2	MFO3	GWO	ALO	GA	PSO
Full Data	Mean Ft.	0.3895	<b>0.3719</b>	0.42313	0.389543	<b>0.3684</b>	0.3888	0.4001
	Best Ft.	0.3414	<b>0.3252</b>	0.3699	<b>0.3170</b>	0.3373	0.3495	0.3699
	Worst Ft.	0.4408	<b>0.4186</b>	0.5243	0.4877	<b>0.41053</b>	0.41057	0.4308
	Std. Ft.	0.029	<b>0.0283</b>	0.047	0.0408	0.0238	<b>0.0175</b>	0.0184
Half Data	Mean Ft.	0.2276	<b>0.2114</b>	0.2718	<b>0.2032</b>	0.2040	0.2273	0.2518
	Best Ft.	0.1692	<b>0.1526</b>	0.1984	<b>0.1384</b>	0.1461	0.1523	0.1923
	Worst Ft.	0.3053	<b>0.2748</b>	0.3893	0.3816	<b>0.2384</b>	0.2977	0.3282
	Std. Ft.	0.0362	<b>0.0349</b>	0.0427	0.0557	<b>0.02813</b>	0.0404	0.0395

Table 2 summarizes the following points by using RF Classifier:

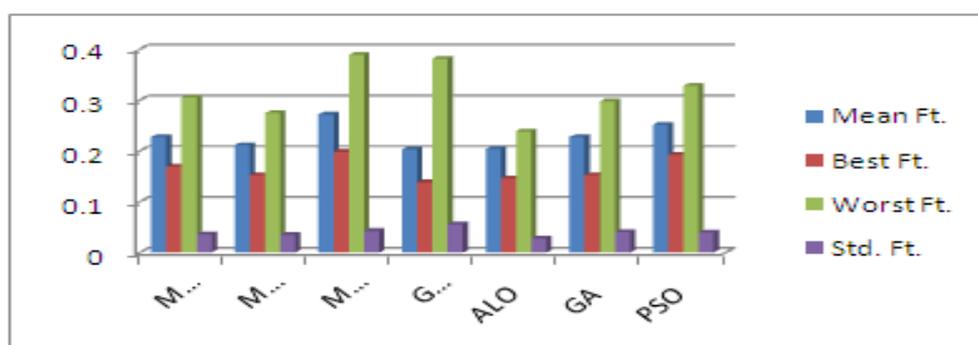
- From the first experiment, we can notice that MFO2 has best mean and best fitness values and very promising with respect to GWO and ALO, while MFO3 performs the best in case of worst and std. fitness values over the original MFO as well as all other optimization algorithms as explained from Figure 10.
- According to the second experiment we can remark that MFO2 produces excellent fitness performance over MFO as well as gives very competitive results with respect to GWO and ALO algorithms while it exceeds all the optimizers' results in case of worst fitness; that is very noticeable from Figure 11.

**TABLE 2:** The Fitness Results of the experiments by using RF Ensemble Classifier

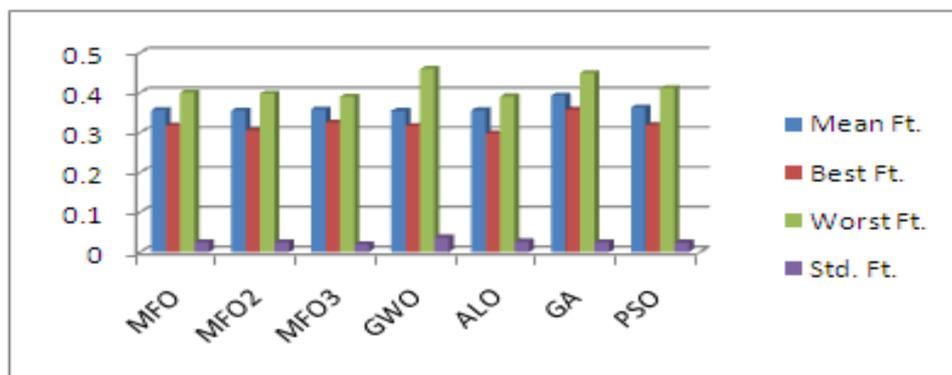
Experiment	Measure	MFO	MFO2	MFO3	GWO	ALO	GA	PSO
Full Data	Mean Ft.	0.3557	<b>0.3545</b>	0.3576	<b>0.3540</b>	0.3553	0.3921	0.3619
	Best Ft.	0.3170	<b>0.3048</b>	0.3252	0.3155	<b>0.2967</b>	0.3565	0.3183
	Worst Ft.	0.4000	0.3975	<b>0.39025</b>	0.4593	0.39028	0.4490	0.4105
	Std. Ft.	0.0248	0.0249	<b>0.01919</b>	0.0363	0.0263	0.0242	0.0238
Half Data	Mean Ft.	0.2004	<b>0.1880</b>	0.2119	<b>0.1876</b>	0.1927	0.2473	0.1977
	Best Ft.	0.1679	<b>0.1307</b>	0.1562	0.1526	<b>0.1615</b>	0.1860	0.1349
	Worst Ft.	0.2519	<b>0.2366</b>	0.3587	0.2442	0.2443	0.2977	0.2519
	Std. Ft.	0.0290	<b>0.0287</b>	0.0502	0.0278	<b>0.0250</b>	0.0339	0.0318



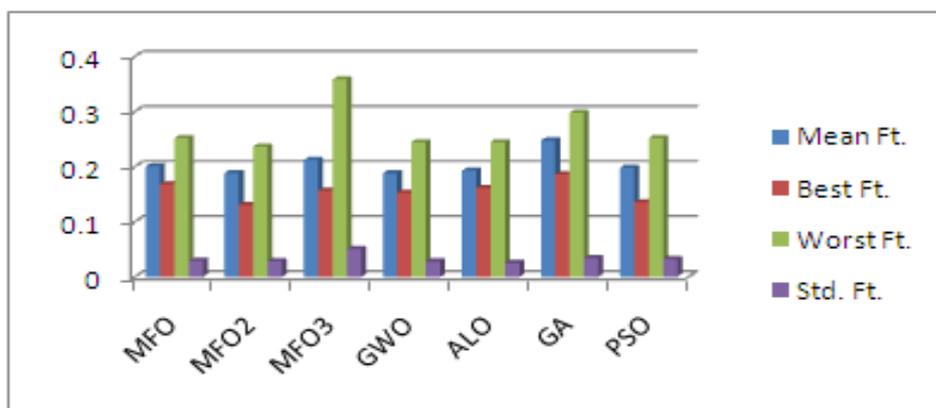
**Figure 8** The Fitness Performance of Exp. I by KNN Classifier



**Figure 9** The Fitness Performance of Exp. II by KNN Classifier



**Figure 10** The Fitness Performance of Exp. I by RFs Ensemble Classifier



**Figure 11** The Fitness Performance of Exp. II by RFs Ensemble Classifier

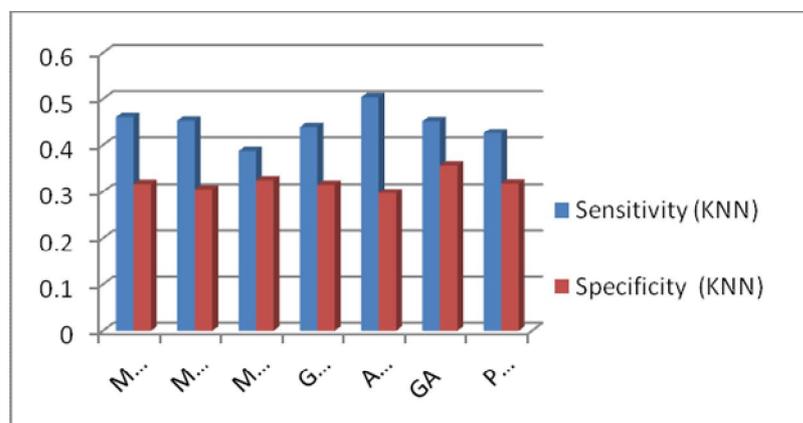
Table 3, summarizes the validity results obtained from conducting the two different experiments from all optimizers we used; as we can see from Exp. I that MFO2 performing well in the sensitivity measure with respect to MFO and ALO, while MFO3 obtained the best specificity result over MFO and MFO2 and is very comparative to the value obtained by GWO. According to Exp. II, we can notice that MFO2 sensitivity results is very near from MFO, and GWO results as well. MFO2 obtains the best specificity value over MFO and MFO3 and is performing very comparative to GWO.

According to Figure 12(A) we can conclude that regardless the classifier used the performance of MFO and MFO2 is very near, while MFO3 is the best in specificity measure over MFO and MFO2 as well as performing comparatively according to GA in case of using KNN;

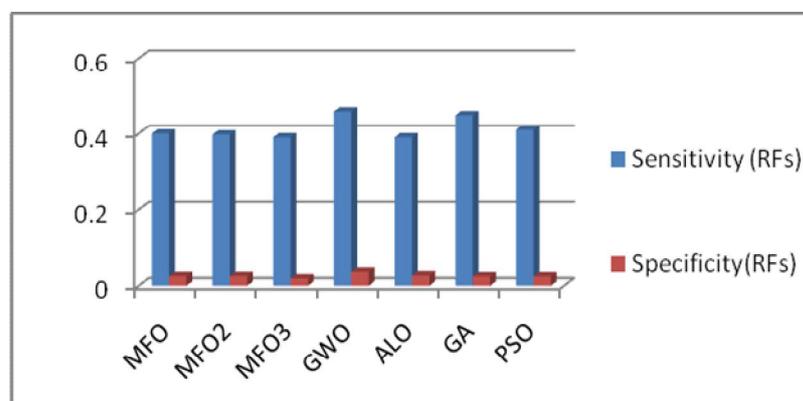
From Figure 12(B) one can observe that MFO2 with RFs classifier performs the best specificity over the other optimizers and behaves very well with respect to GWO. From Figure12 (A) and (B) the evidence that proves the validity of the test by using MFO2 and MFO3.

**TABLE 3:** Sensitivity and Specificity measures for the optimizers from KNN and RF Classifiers by Exp. I

Experiment	Measure	MFO	MFO2	MFO3	GWO	ALO	GA	PSO
Full Data	Sensitivity (KNN)	<b>0.4618</b>	0.4543	0.3873	0.4403	<b>0.5039</b>	0.4528	0.4263
	Specificity (KNN)	0.3170	0.3048	<b>0.3252</b>	0.3155	0.2967	<b>0.3565</b>	0.3183
	Sensitivity (RFs)	<b>0.4000</b>	0.3975	0.39025	<b>0.4593</b>	0.39028	0.4490	0.4105
	Specificity (RFs)	0.0248	<b>0.0249</b>	0.01919	<b>0.0363</b>	0.0263	0.0242	0.0238



**Figure 12(A)** Validity Measures' of Exp. I by KNN Classifier.



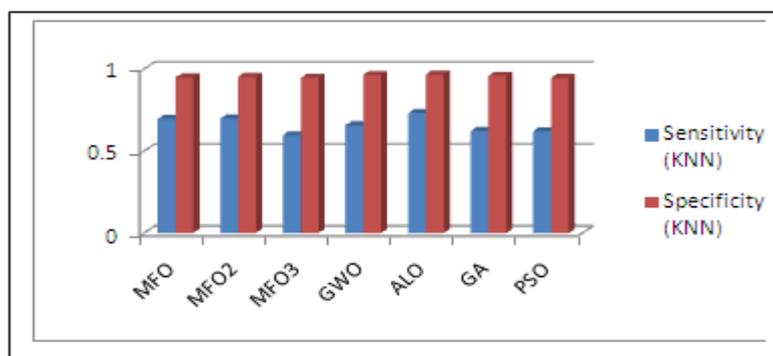
**Figure 12(B)** Validity Measures' of Exp. I by RFs Classifier.

**TABLE 4:** Sensitivity and Specificity Measures for the optimizers from KNN and RFs Classifiers by Exp. II

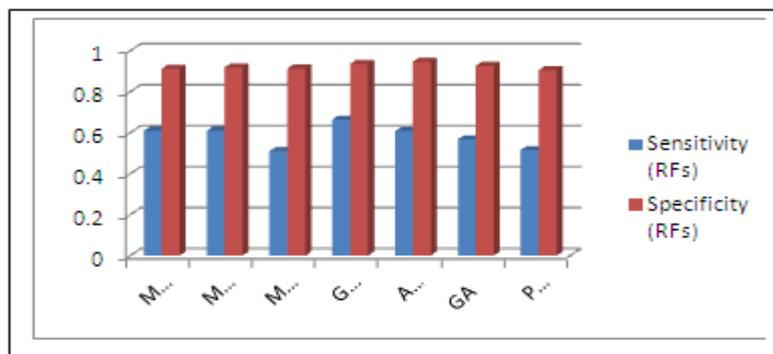
Experiment	Measure	MFO	MFO2	MFO3	GWO	ALO	GA	PSO
Half Data	Sensitivity (KNN)	0.6953	<b>0.6973</b>	0.5954	0.6561	<b>0.7302</b>	0.6222	0.6197
	Specificity (KNN)	0.9477	<b>0.9528</b>	0.9460	0.9628	<b>0.9664</b>	0.9571	0.9450
	Sensitivity (RFs)	0.6100	<b>0.6101</b>	0.5104	<b>0.6637</b>	0.6084	0.5677	0.5156
	Specificity (RFs)	0.9121	<b>0.9185</b>	0.9132	0.9363	<b>0.9476</b>	0.9263	0.9057

Table 4 summarizes the following points:

- MFO2 behaves efficiently in the measures of validity regardless the type of classification algorithm used over MFO and is comparative to ALO results in case of using KNN classifier.
- MFO proves its superiority over original MFO in case of testing the specificity and still produces efficient and promising results with respect to GWO and ALO optimizers.



**Figure 13(A)** The Validity Measures of Exp. II by KNN Classifier.



**Figure 13(B)** The Validity Measures of Exp. II by RFs Classifier.

Figures 13 (A), (B) shows how MFO2 is efficient and outperforms the original MFO in the validity measures as well as is very promising algorithm with respect to other optimizers. Also we can observe that using RFs ensemble classifier in our prediction system is more efficient in prediction and the classification process than using KNN classifier.

## 6. CONCLUSIONS AND FUTURE WORK

The paper proposed two new modified versions of MFO algorithm based mainly on trying different spiral flying path of moths around the artificial lights (Flames). The mathematical framework of the new versions of MFO algorithm combining into a hybrid ensemble classification algorithm based on MFO2-RF with the help of Wrapper feature selection approach that can be used in the prediction of terrorist groups among different regions and countries. The proposed model implements the novel MFO versions and wrapper feature selection approach in order to select optimal feature subset for classification process based on random forests (RFs) ensemble classifier to improve and enhance the classification accuracy while minimizing the number of selected features.

The performance of the hybrid MFO2-RFs and MFO3-RFs models are tested via two different experiments during 20 iterations and the results are benchmarked for evaluation and comparison with the original MFO, as well as a number of different popular optimization algorithms such as Ant Lion Optimizer (ALO), Grey Wolf Optimization (GWO) algorithm, Particle Swarm Optimization (PSO) and genetic algorithm (GA), another direction of comparison is by comparing the obtained results of the hybrid model based on RF ensemble classifier with another well known classifier as K-nearest neighbor (KNN).

A set of assessment performance indicators are used to analyze, evaluate and compare between the obtained results which prove the capability of the proposed versions of MFO (MFO2, MFO3) algorithms to search the feature space for the optimal feature combination as well as enhancing the classification accuracy compared to other well-known conventional, heuristics and meta-heuristics search algorithms. Experimental results of the proposed modified versions of MFO provide very promising and competitive performance in reaching global minima, as well as prove good performance of MFO2, MFO3 and an advance over the original MFO algorithm with high stability, validity, and robustness results over other searching methods.

Further investigation on the parameters values and testing the proposed hybrid MFO2-RF and MFO3-RF algorithms with other feature selection approaches on different dimensions data sets are different and various areas for future research.

## References

- [1] S. Mirjalili, "Moth-flame optimization algorithm: A Novel Nature-Inspired Heuristic Paradigm", Elsevier, Knowledge-Based Systems, 2015.
- [2] J. H. Holland, "Genetic algorithms, Sci. Am. 267", pp.66–72, 1992.
- [3] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.
- [4] A. Coloni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: Proceedings of the First European Conference on Artificial Life, pp. 134–142, 1991.
- [5] R. Storn, K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", J. Glob. Opt. 11, pp. 341–359, 1997.
- [6] I. Rechenberg, Evolution strategy: optimization of technical systems by means of biological evolution, Fromman-Holzboog, Stuttgart 104, 1973.
- [7] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial intelligence through simulated evolution, 1966.
- [8] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
- [9] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
- [10] Bowser, A. Edward. An Elementary Treatise on Analytic Geometry: Embracing Plane Geometry and an Introduction to Geometry of Three Dimensions (4th ed.), D. Van Nostrand, 1880.

## AUTHORS



**Ghada M. A. Soliman** received her degree in B. of science in Operations Research and Decision Support, at the faculty of Computers and Information, Cairo University in 2001. Then she received her Master Degree in Operations Research in 2007, she is currently a PhD student. Her employment experience includes FCI institution since 2001; she is employed with the Department of Operations Research as Lecturer Assistant. Her research interests include modeling, simulation, data mining and soft computing.



**Prof. Dr. Motaz M. H. Khorshid**, is a Professor of Operations Research, - Department of O.R. & D. S. - Faculty of Computers Information-Cairo University. His research interest is focused on computer aided Modeling and Decision Support Systems (DSS), Computer Simulation Methodology, Economy-wide Modeling and Analysis, Socioeconomic Data Systems, Planning and Management of Higher Education and Scientific Research, Development Planning, Quantitative Management and Corporate Planning Models.



**Dr. Tarek H. M. Abou-El-Enien** is an Associate Professor of Operations Research, - Department of O.R. & D. S. - Faculty of Computers & Information- Cairo University. His research interests in decision analysis, soft computing, modeling, data mining and multi-level multi-objective Programming field.