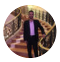
A Hybrid Moth-Flame Optimization and Extreme Learning Machine Model for Financial Forecasting

Conference Paper · March 2018

3 authors:


Mustafa Abdul Salam
Benha University


Eid Emary
Cairo University


Hossam M. Zawbaa
Beni Suef University

A Hybrid Moth-Flame Optimization and Extreme Learning Machine Model for Financial Forecasting

Mustafa Abdul Salam[1], Eid Emary[2,3], Hossam M. Zawbaa[4,5]
[1]*Faculty of Computers and Information, Benha University, Egypt*
[2]*Faculty of Computers and Information, Cairo University, Egypt*
[3]*Faculty of Computer Studies, Arab Open University, Egypt*
[4]*Faculty of Computers and Information, Beni-Suef University, Egypt*
[5]*Faculty of Mathematics and Computer Science, Babes-Bolyai University, Romania*

***Abstract*—In this paper, a system for stock market prediction is proposed based on a hybrid moth-flame optimizer (MFO) and extreme learning machine (ELM). ELM is a recent amazing model for function approximation and classification applications. ELM takes small training time, but it needs a very large number of nodes in the hidden layer. This huge number of nodes increases evaluation time of the network, and there is no guarantee of optimality of weights and biases on the hidden layer. MFO is a novel optimization algorithm that imitates the moving behaviour of moths. MFO is used to minimize the number of neurones in the hidden layer to enhance the performance of the ELM model while ensuring optimality by the appropriate selection of weights and biases. The proposed MFO-ELM model was applied to monthly stock market prediction where a set of financial technical indicators were used with the proposed model to enhance its generalization ability and prediction accuracy. The proposed model was tested with different data sets representing the various sectors in Standard and Poor 500 stock market index. The MFO algorithm was compared against flower pollination algorithm (FPA) in optimizing weights and biases of ELM model. Simulation results showed that MFO enhanced the prediction accuracy of ELM, and gave better accuracy than FPA and standard ELM model.**

***Index Terms*—Extreme Learning Machine, Moth-Flame Optimization Algorithm, Flower Pollination Algorithm, Stock market prediction.**

I. Introduction

Artificial neural networks are considered one of the most common machine learning models used in regression and classification domains [1]. The learning algorithm is considered the cornerstone of the neural network. Classical gradient-based learning algorithms such as Levenberg Marquardt (LM) and Scaled Conjugate Gradient (SCG) are suffering from overfitting, local minima, and they consume a long time to learn [2], [3], [4].

Extreme learning machine (ELM) was introduced by Huang [5] to overcome the problems found in gradient-based learning algorithms. ELM is used as a supervised learning method for SLFN method. ELM has high accuracy and fast prediction speed while solving numerous real-life problems [2], [6]. ELM randomly selects the input weights and hidden layer biases instead of fully tuning all the internal parameters such as gradient-based algorithms. ELM could analytically determine the output weights [2]. Due to random choosing of input weights and hidden layer biases, ELM needs more hidden neurons than gradient-based learning algorithms [7], [8].

The bio-inspired algorithms were used in optimizing ELM to overcome its drawbacks. In [7] differential evolution (DE) algorithm was applied to select input weights and biases to determine the output weights of ELM. DE-ELM achieved good generalization performance with a compact structure. In [9] DE-ELM was used for the classification of hyperspectral images, and it improved classification accuracy and computation time. In [10] Evolutionary ELM based on PSO algorithm is proposed, and PSO algorithm improved the performance of traditional ELM. In [11] a new method combined ELM with an improved PSO called is proposed to improve the convergence performance of ELM. In [12] an evolutionary approach for constituting ELM ensembles is introduced to direct the choice of base learners and provide an optimal solution. In [13] Genetic ensemble of ELM is introduced. In [14], a new, real-coded genetic algorithm approach called 'RCGA-ELM' is proposed to select the optimal number of hidden neurons, input weights and bias values of ELM model which results in better performance. In [15] ELM based genetic algorithm was applied in power system economic dispatch. In [16], a recent Flower Pollination Algorithm (FPA) is proposed to optimize ELM model and achieved better results compared to classical ELM model in stock market prediction. In [17], a hybrid dragonfly algorithm with an extreme learning machine for prediction is presented, in this model, dragonfly optimizer enhanced the performance of ELM model in the field of prediction. In [18], a feature selection approach based on moth-flame optimization (MFO) algorithm is introduced, and MFO algorithm is outperformed compared methods.

In this paper, a new proposed model (MFO-ELM) integrating ELM with a very recent evolutionary algorithm called Moth-Flame Optimization Algorithm (MFO) is proposed and applied to stock price prediction. MFO [19] is proposed to optimize the input weights and hidden biases of ELM. The proposed model is also combined with six financial technical indicators computed from stock market data to enhance the prediction accuracy and generalization performance of the proposed model. The rest of this paper is organized as follows:

978-1-5386-0879-1/18/$31.00 ©2018 IEEE

Section II presents the background information of Extreme Learning Machine (ELM) model and Moth-Flame Optimization (MFO) algorithm. Section III is devoted to the proposed model and its implementation for the stock price prediction. The experimental results are explained in section IV while the main conclusion of the proposed model is presented in Section V.

II. Preliminaries

A. Extreme Learning Machine (ELM)

Extreme learning machine (ELM) model has been proposed for single hidden layer feed-forward neural networks (SLFNs). In ELM model, the connections between the input layer and the hidden neurons are randomly selected and remain unchanged during the learning process [7]. The output connections are then tuned via minimizing the cost function through a linear system [20].

Suppose we are training SLFNs with N hidden neurons and activation function function $f(x)$ to learn M distinct samples(x_i, t_i),

where: $x_i = [x_{i1}, x_{i2}, ..., x_{ik}]^T \in R^k$,$t_i = [t_{i1}, t_{i2}, ..., t_i d]^T \in R^d$. By doing so, the nonlinear system has been converted to a linear system:

$$H\beta = T, \tag{1}$$

where H is the hidden-layer output matrix denoted by:

$$H = \begin{bmatrix} f(w_1.x_1 + b_1) & \dots & f(w_N.x_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(w_1.x_M + b_1) & \dots & f(w_N.x_M + b_N), \end{bmatrix} \tag{2}$$

where $w_j = [w_{j1}, w_{j2}, ..., w_{jk}]^T, (j = 1, 2, ..., N)$ is the weight vector connecting j_{th} hidden neuron and input neurons, and b_j denotes the bias of j_{th} hidden neuron; $w_j.x_i (i = 1, ..., M)$ denotes the inner product of w_j and x_i; $\beta = [\beta_1, \beta_2, ..., \beta_N]^T$, the matrix of output weights and $\beta_j = [\beta_j 1, \beta_j 2, ..., \beta_j d]^T$, $(j = 1, 2, ..., N)$ denotes the weight vector connecting the j_{th} hidden neuron and output neurons; $T = [\hat{t_1}, \hat{t_2}, ..., \hat{t_M}]^T$ is the matrix of targets (desired output).

In the case where the $SLFN$ perfectly approximates the data, the errors between the estimated outputs $\hat{t_i}$ and the actual outputs t_i are zero and the relation is:

$$t_i = \sum_{j=1}^{N} \beta_j f(w_j.x_i + b_j), \tag{3}$$

The output weights are determined by the least square solution to the represented model in equation (4) and the minimum norm least square (LS) solution to the linear system is:

$$\hat{\beta} = H * T, \tag{4}$$

where $H*$ is the Moore-Penrose (MP) which generalized inverse of matrix H. The minimum norm LS solution is unique and has the lowest norm among all the LS solutions. ELM using such MP inverse method tends to obtain good generalization performance with dramatically increased learning speed [2].

B. Moth-Flame Optimization (MFO) Algorithm

Moth-Flame Optimization (MFO) was introduced by Seyedali Mirjalili in 2015 [19]. MFO is inspired by the navigation approach of Moths. Moths have evolved to fly in the night using the moonlight. They rely on a transverse orientation navigation. In this method, a moth flies by maintaining a fixed angle with respect to the moon. This approach is considered a very efficient method for travelling long distances in a straight path [21], [22]. Since the moon is far away from the moth, this method guarantees flying in straight line [22]. An algorithm inspired by this type of motion is presented in the following section.

Moths and flames are the main components of the artificial algorithm. The candidate solutions are moths.The moth's positions in the search space are the variables of the problem. Therefore, moths can fly in one, two, or hyper-dimensional space with varying location vectors. The MFO optimizer is a population-based algorithm, and moths are employed as search agents in the problem space.

Flames are considered the best positions of moths that are computed so far. Each moth searches around a flame and updates it if a better solution is found. Therefore, flames are also d-dimensional data points. Given logarithmic spiral, a given moth updates its position with respect to a given flame as follows:

$$S(M_i, F_j) = D_i.e^{bt}.cos(2\pi t) + F_j \tag{5}$$

where D_i refers to the Euclidian distance of the $i - th$ moth for $thej - th$ flame, b is a constant for determining the shape of the logarithmic spiral, M_i indicates the $i - th$ moth, F_j indicates the $j - th$ flame and t is a random number in range $[-1, 1]$.

The next position of a moth is determined to the corresponding flame. The t parameter in the spiral formula specifies how much the coming location of the moth should be near to the flame. Therefore, a hyper-ellipse can be expected around the flame in all directions, and the next position of the moth would be in this space. The spiral formula makes a moth fly around a flame, and allowing for both exploration and exploitation of the solutions.

We assume that t is a random number in $[r, 1]$ where r is linearly decreased from $-1 to - 2$ over the course of iteration and is called convergence constant. With this approach, moths head for exploiting their corresponding flames more preciously proportional to the number of iterations.

To enhance the probability of converging to a global solution, a given moth is obliged to update its position using one flame. In every step or iteration and after updating the list of flames, the flames are sorted according to their fitness function values. The moths then update their positions based on their corresponding flames.

To allow for much exploitation of the best promising solutions, the number of flames to be followed is decreased concerning the iteration number as follows:

$$N_{flames} = round(N - l * \frac{N-1}{T}) \tag{6}$$

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-ELM Model

Extreme learning machines have the advantage of low training time while keeping acceptable classification and regression performance on the condition that a huge number of hidden nodes are selected in the model. The huge number of nodes in the hidden layer slows down the testing performance of ELM while there is no grantee of optimality of the set of weights on the hidden layer. Therefore, in proposed model less number of nodes in the hidden layer is selected to speed up the performance of the ELM while ensuring optimality by the appropriate selection of hidden layer weights and biases. The proposed model still exploiting the same strategy for setting the weights and biases of the output layer.

MFO algorithm is used to select the weights and biases of the hidden layer that maximizes the overall performance of the ELM. The number of variables to be estimated by the MFO is M variables with $M = (N_H + 1) * N_I$ where N_H and N_I are the numbers of nodes in the hidden and input layers in order. The range of individual variables is set to the range [-1, 1]. The used fitness function is:

$$f(w) = \sum_{j=1}^{N_{Validation}} \sum_{i=1}^{N_O} \mid O_{ij} - d_{ij} \mid \tag{7}$$

where $N_{Validation}$ is the number of data points in the validation set, N_O is the number of nodes in the output layer. O_{ij} is the actual output on node i as a result of applying sample j to the ELM, and d_{ij} is the desired output on node i for a sample j. An algorithm depicting the proposed MFO-ELM model is presented in algorithm 1. We can briefly describe the main components of the algorithm as follows:

1) Initialization: a set of moths positions with each position representing a neural's hidden layer weights and biases and ranges from [-1, 1].
2) Moth position evaluation: a moth position represents a whole set of weights and biases for the hidden layer and hence to calculate its fitness the whole neural network is constructed given the hidden layer weights and biases and hence the output layer weights and biases are calculated using the MP generalized inverse matrix given the training data. Hence, the fitness of the agent calculated as sum square error calculated over the validation data set.
3) Moth repositioning: Moths updates their positions according to the native MFO algorithm given the calculated fitness and the current flames.

input : n Number of moths (search agents)
    N The maximum number of flames
    T Number of iterations
    Trn Training data set
    Vld Validation data set
output: f_{best} Optimal hidden weights and biases
    $f(f_{best})$ Sum square error for the NN over
the validation set f_{best}

1) Initialize population of n flames positions randomly with hidden layers weight and biases.
2) **while** *Stopping criteria not met* **do**
    Update the No. of flames to be used N_{flames} according to equation 6.
    Calculate the fitness of all the n moths by
    foreach $Moth_i$ *with* $i \leq n$ **do**
        a) construct NN given the hidden layer weights and biases of $Moth_i$
        b) Calculate the output layer weights using the MP matrix given training data Trn and hidden layer weights and biases.
        c) Evaluate NN model using the validation data Vld
    end
    if *first iteration* **then**
        Sort the moths according to their fitness and place the result in the flame matrix.
    else
        Combine the population of past moths and flames. Sort the combined population from best to worst. Select the best N positions from the sorted combined population as the flames.
    end
    Calculate the convergence constant r
    foreach $Moth_i$ *with* $i \leq n$ **do**
        Calculate t as $t = (r-1) * rand + 1$
        if $i \leq N$ **then**
            Update $Moth_i$ position according to $Flame_i$ using equation 5
        else
            Update $Moth_i$ position according to $Flame_{N_{flames}}$ using equation 5
        end
    end
end

Algorithm 1: The hybrid MFO-ELM for regression

A. MFO-ELM Applied for Stock market prediction

The proposed MFO-ELM model is applied to the stock market prediction to assess its generality and prediction accuracy in this stream time series data. The stock market data sets contain (High, Low, Open, Close, and Volume). A set of financial indicators are calculated from the data sets and are used as inputs to the proposed model. Six financial, technical indicators are calculated from the data sets to enhance prediction accuracy and generalization performance of stock prediction. The financial, technical indicators used are:

1) **Price Momentum Oscillator (PMO)** [23]: an oscillator based on a Rate of Change (ROC) estimation that is exponentially smoothed twice. Since the PMO is normalized, it can also be used as a relative strength tool. Their PMO value can thus rank stocks as an expression of relative strength. TC = todayâĂŹs close price, and $TDAC$ = close price ten days ago. The PMO is calculate as follows:

$$PMO = TC - TDAC, \tag{8}$$

2) **Relative Strength Index (RSI)** [24]: It tries to specifies over-sold and over-bought situations of an asset. The equation for calculating RSI is as follows:

$$RSI = 100 - [100/(1 + RS)], \tag{9}$$

where RS is the average of x days up closes divided by average of x days down closes.

3) **Money Flow Index (MFI)** [25]: This measures the strength of money in and out of a security. The formula for MFI is as follows:

$$MF = TP * V, \tag{10}$$

where TP is typical price, and V is money volume. Money Ratio (MR) is calculated as:

$$MR = (PositiveMF/NegativeMF), \tag{11}$$

$$MFI = 100(100/(1 + MR)), \tag{12}$$

4) **Exponential Moving Average (EMA)** [26]: This indicator returns the exponential moving average of a field over a given period of time. EMA formula is as follows:

$$EMA = [\alpha * TC] + [1 - \alpha * YE], \tag{13}$$

where TC is today close and YE is Yesterday EMA.

5) **Stochastic Oscillator (SO)** [27]: The stochastic oscillator is defined as a measure of the difference between the current closing price of a security and its lowest low price, about its highest high price for a given period. The equation for this calculation is as follows:

$$SO = [(CP - LP)/(HP - LP)] * 100, \tag{14}$$

where CP is Close price, LP is Lowest price, HP is Highest Price, and LP is Lowest Price.

6) **Moving Average Convergence / Divergence (MACD)** [28]: This function calculates difference between a short and a long term moving average for a field. The formulas for calculating MACD and its signal as follows:

$$MACD = [0.075 * E] - [0.15 * E], \tag{15}$$

where E is EMA(CP).

The set of financial indicators for the given historical period are fed as an input to the regression model given the target as the current stock market value.

IV. Experimental Results

A. Stock Market Datasets

The proposed MFO-ELM model was trained and tested with five monthly datasets covering different sectors in Standard and Poor 500 stock market index (S&P500). The datasets were selected in the various sectors to test the generalization ability and accuracy of the proposed model since each sector changes and fluctuations are different.

The five stock market datasets are namely: American Airlines, HP, Yahoo, Toyota, and Walt Disney companies. Datasets periods are from Jan. 2000 to Jan. 2014, which are divided into two parts (training part 70% and testing part 30%). All datasets are available in [29].

B. Parameters Settings

The proposed and compared models were trained with one thousand iterations. In MFO algorithm, the number of moths is thirty that is equal to the number of flames. In FPA algorithm, the number of flowers is thirty. In FPA algorithm, Local pollination and global pollination is controlled by a switching probability that is 0.8 which was selected after several trials. ELM has seven nodes in input layer representing the six technical indicators and monthly close price. It has one hundred hidden nodes since it needs more hidden nodes than classical gradient descent training algorithms. It has one node in output layer representing the next week close price.

The parameters settings of MFO, FPA, and ELM algorithms are summarized in Table(I).

TABLE I: MFO algorithm parameters

Algorithm	Parameter	Value
MFO	No. of Moths	30
	No. of Flames	30
	No. of Iterations	1000
FPA	No. of Flowers	30
	No. of Iterations	1000
	Probability switch	0.8
ELM	Input nodes	7
	Hidden nodes	100
	Activation fun.	Sigmoid
	Output nodes	1
	No. of Iterations	1000

C. Performance Evaluation Criteria

Proposed and compared models were evaluated according to three evaluation criteria. These criteria measure the prediction accuracy and the price trend or direction. The evaluation criteria are calculated as follows:

1) Root Mean Square Error (RMSE)

$$RMSE = \sqrt{(\frac{1}{n}\sum_{i=1}^{n}(A_i - F_i)^2)} \qquad (16)$$

2) Symmetric Mean Absolute Percentage Error (SMAPE)

$$SMAPE = \frac{\sum_{i=1}^{n}|A_i - F_i|}{\sum_{i=1}^{n} A_i + F_i} \qquad (17)$$

D. Simulation Results analysis

Figures from figure 1 to figure 5 outline the application of Proposed MFO-ELM model on test datasets for different companies. Figures (1, 2, and 3) describe results of three companies (American Airlines, HP, and Yahoo) which are in different stock sectors. The proposed MFO-ELM model could easily cope with fluctuations in stock time series data. MFO-ELM model achieved the lowest error, followed by FPA-ELM model. ELM and LM models fall into local minima and over-fitting problems and have the worst error. Figure 4 outlines the results of Toyota datasets. From the figure, we can notice that MFO-ELM model has the closest to actual with a little advance over others compared models since time series have no fluctuations and the datasets are normal.

Figure 5 introduces the results of Walt Disney datasets. The first half of the time series is normal while the other half has fluctuations. Therefore, the advance of the proposed model is clear in the second half of this datasets. ELM and LM model still the worst. Table(II) and table(III) represent the RMSE and SMAPE errors values regarding proposed MFO-ELM model, and compared models. Form table (II), and table(III), we noticed that the proposed model achieved the lowest error value over all datasets, followed by FPA-ELM model, while ELM and LM models could not cope with fluctuations of datasets.

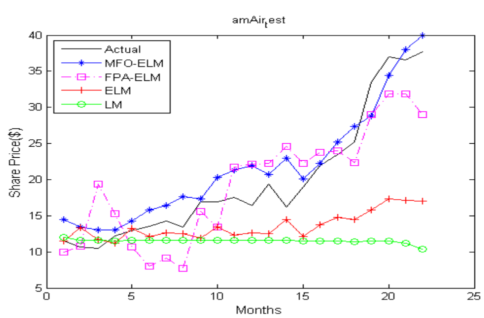


Fig. 1: American Airlines test results.

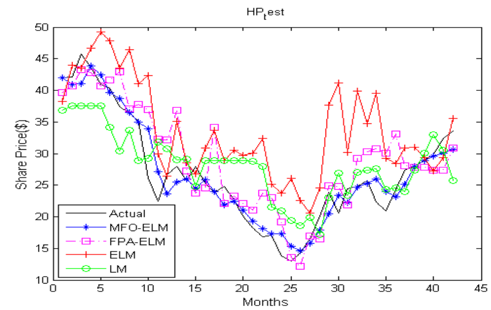


Fig. 2: HP test results.

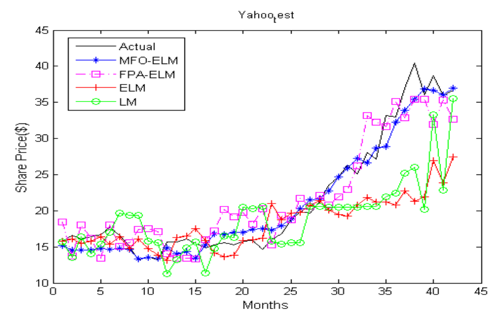


Fig. 3: Yahoo test results.

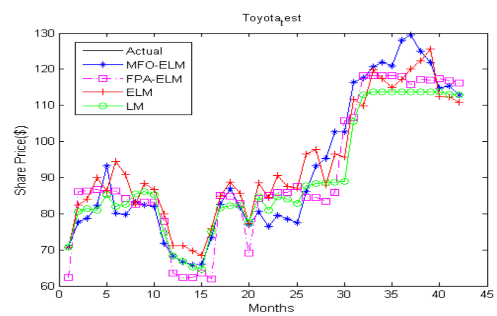


Fig. 4: Toyota test results.

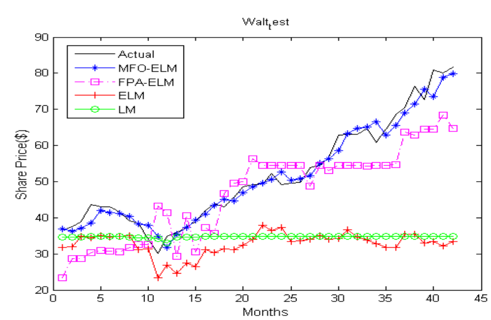


Fig. 5: Walt Disney test results.

TABLE II: RMSE of Proposed MFO-ELM and compared models

Datasets	MFO-ELM	FPA-ELM	ELM	LM
American Air.	3.03	4.75	9.42	12.14
HP	2.41	4.86	8.74	5.27
Yahoo	1.77	3.08	6.32	5.88
Toyota	0.01	6.81	6.47	6.32
Walt Disney	2.57	8.92	22.98	21.92

TABLE III: SMAPE of Proposed MFO-ELM and compared models

Datasets	MFO-ELM	FPA-ELM	ELM	LM
American Air.	0.06	0.10	0.20	0.27
HP	0.03	0.07	0.12	0.08
Yahoo	0.03	0.06	0.10	0.11
Toyota	0	0.03	0.03	0.03
Walt Disney	0.02	0.08	0.21	0.19

V. Conclusion and Future work

In this paper, a recent bio-inspired Moth-Flame Optimization (MFO) algorithm is proposed to optimize Extreme Learning Machine (ELM) model. MFO was used to choose the input weights optimally and hidden layer biases to make network structure more compact, instead of random choosing found in traditional ELM model. The Proposed MFO algorithm is compared with flower pollination algorithm (FPA) in optimizing ELM model. The Proposed MFO-ELM model is applied to monthly stock price prediction. Technical indicators were used with the proposed model to enhance the prediction accuracy of the model. The proposed model convergence to a global minimum can be expected in little iterations followed by FPA-ELM model. The proposed model overcame the over-fitting and local minima problems, which found in ELM and LM models. MFO-ELM model is better than FPA-ELM model especially in the case of fluctuations in stock time series. MFO-ELM model parameters are few and can be tuned easily. MFO-ELM achieved the lowest error value for all compared evaluation criterion (RMSE, and SMAPE) followed by FPA-ELM model while ELM and LM algorithms were the worst. MFO is very promising in optimizing ELM model, and more research efforts should be devoted to this exciting area.

References

[1] P. Bas O. Simula C. Jutten A. Lendasse Y. Miche, A. Sorjamaa. Op-elm: Optimally pruned extreme learning machine. *IEEE Trans. Neural Networks*, 21(1):158–162, 2010.

[2] C.K. Siew G.B. Huang, Q.Y. Zhu. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[3] D.S. Huang F. Han. Improved extreme learning machine for function approximation by encoding a priori information. *Neurocomputing*, 69(1):2369–2373, 2006.

[4] D.S. Huang. A constructive approach for finding arbitrary roots of polynomials by neural networks. *IEEE Trans. Neural Networks*, 15(2):477–491, 2004.

[5] C.K. Siew G.B. Huang, Q.Y. Zhu. Extreme learning machine: a new learning scheme of feedforward neural networks. In *International Joint Conference on Neural Networks (IJCNN'2004)*, pages 985–990. Budapest, Hungary, 2004.

[6] R. Wang Y. Cai J. Cao F. Wang H. Min X. Deng X. Li, H. Xie. Empirical analysis: stock market prediction via extreme learning machine. *Neural Comput and Applic*, 1(3):1âĂŞ12, 2014.

[7] P.N. Suganthan G.B. Huang Q.Y. Zhu, A.K. Qin. Evolutionary extreme learning machine. *Pattern Recognition*, 38(10):1759–1763, 2005.

[8] C.Y. Miao Z.H. Man G.P. Zhao, Z.Q. hen. On improving the conditioning of extreme learning machine: a linear case. In *7th International Conference on Information, Communications and Signal Processing(ICICS2009)*, page 1âĂŞ5. Maucu, China, 2009.

[9] F. Melgani Y. Bazi, N.Alajlan. Differential evolution extreme learning machine for the classification of hyperspectral images. *Geoscience and Remote Sensing Letters*, 11(6):1066–1070, 2014.

[10] Y. Shu Y. Xu. Evolutionary extreme learning machine-based on particle swarm optimization. *International Symposium on Neural Networks*, 3971(1):644–652, 2006.

[11] Q. HuaLing F. Han, H. FenYao. An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing*, 116(1):87–93, 2013.

[12] M. Alhamdoosh D. Wang. Evolutionary extreme learning machine ensembles with size control. *Neurocomputing*, 102(1):98–110, 2013.

[13] Z. Wu J. Yang X. Xue, M. Yao. Genetic ensemble of extreme learning machine. *Neurocomputing*, 129(1):175–184, 2014.

[14] N. Sundararajanc S. Suresha, S. Saraswathib. Performance enhancement of extreme learning machine for multi-category sparse data classification problems. *Engineering Applications of Artificial Intelligence*, 23(7):1149–1157, 2010.

[15] J. Zhao Z. Dong H. Yang, J. Yia. Extreme learning machine based genetic algorithm and its application in power system economic dispatch. *Neurocomputing*, 102(7):154–162, 2013.

[16] M. Abdul Salam O. Hegazy, O.S. Soliman. Fpa-elm model for stock market prediction. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(2):1050–1063, 2015.

[17] E. Emary K.K.A. Ghany B. Parv M.A. Salam, H.M. Zawbaa. A hybrid dragonfly algorithm with extreme learning machine for prediction. *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6, 2016.

[18] B. Parv M. Sharawi H.M. Zawbaa, E. Emary. Feature selection approach based on moth-flame optimization algorithm. *IEEE Congress on Evolutionary Computation (CEC)*, pages 4612–4617, 2016.

[19] S. Mirjalili. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89(1):1050–1063, 2015.

[20] L. Zhiping C. Jiuwen. Extreme learning machines on high dimensional and large data applications: A survey. *Mathematical Problems in Engineering*, 2015(1):1–13, 2015.

[21] T.W. Davies J. Hopkins K.J. Gaston, J. Bennie. The ecological impacts of nighttime light pollution:a mechanistic appraisal. *Biological reviews*, 88(1):912–927, 2013.

[22] T. Longcore K.D. Frank, C. Rich. Effects of artificial night lighting on moths. *Ecological consequences of artificial night lighting*, 2015(1):305–344, 2006.

[23] Chartschool, price momentum oscillator (pmo). StockCharts.com, Last access 20 November 2015.

[24] Chartschool, relative strength index (rsi). StockCharts.com, Last access 20 November 2015.

[25] Chartschool, money flow index (mfi). StockCharts.com, Last access 20 November 2015.

[26] Chartschool, moving averages - simple and exponential. StockCharts.com, Last access 20 November 2015.

[27] N.J. Hoboken. A complete guide to technical trading tactics: How to profit using pivot points, candlesticks and other indicators. pages 144–145. Wiley, ISBN 0-471-58455-X, 2004.

[28] Chartschool, moving average convergence divergence (macd). StockCharts.com, Last access 20 November 2015.

[29] *http://finance.yahoo.com*. Yahoo, last accessed 20 November, 2015.