

Long term load forecasting based on hybrid model of feed forward net and modified grey wolf optimization

RIZK M. RIZK-ALLAH¹, I. M. EL-DESOKY¹, A. N. AYAD¹

¹Basic Engineering Science Department
¹Faculty of Engineering, Menofia University
 Shebin El-Kom, EGYPT

Abstract: - This paper presents a hybrid modified grey wolf optimization (MGWO) algorithm with the feed forward net (FFN), named MGWO-FFN, for solving electrical load forecasting. The proposed model is implemented with two stages: firstly, MGWO algorithm estimates the optimum variables of the FFN through the pre-determined training samples. Then the adapted FFN is tested with the remaining other samples and is utilized to predict the electrical peak load (PL). The proposed algorithm is investigated on two real cases (i.e. predicting the annual total electrical load consumption of Beijing's city and the annual PL consumed in Egypt). To prove the superiority of the proposed algorithm, MGWO is validated by comparing with algorithm including classical GWO and PSO algorithms. Both of Beijing's and Egypt's cases results indicate that the proposed MGWO-FFN algorithm outperforms the others where less mean square error (MSE) and more accuracy are obtained compared to the error that yields using the other two algorithms.

Key-Words: - Power Systems, Grey wolf, Forecasting, Optimization, Feed forward net.

Received: January 14, 2020. Revised: July 19, 2020. Accepted: July 26, 2020. Published: August 2, 2020

Abbreviations:

MLP	Multi-layer perceptron	FFN	Feed forward net
GRNN	Generalized regression neural network	GWO	Grey wolf optimizer
MGWO	Modified grey wolf optimizer	PSO	Particle swarm optimizer
LTLF	Long term load forecasting	MTLF	Medium term load forecasting
STLF	Short term load forecasting	PL	Peak load
SALSSVM	Support vector machine with simulated annual algorithm	OLS-LR	Ordinary least square linear regression

1 Introduction

Electricity is recently considered to be the locomotive of progress of all nations and it is the best friend for environment. At nowadays, the need for electric power has increased in all fields. Electricity becomes the basis of economic growth, community development, industrial production and everything related to people, so that governments all over the world attach a great importance to generate electrical power with less harmful and more effective ways. No effort is spared for the issue of finding electrical power sources, constructing power plants and developing all power system network parts to fulfill a strong smart grid for the security, reliability and stability of the operation of electric power grid. Long term electric load forecasting plays a vital role to electricity systems; it leads to reliable and sustainable electrical power systems by telling us in the future how will we can face the challenges of the electricity consumption using developed smart grid and additional electrical power

plants. Because of the non-linearity of the load curve relationships the forecasting process is very difficult and exhausting for researchers. So researchers and participants seek to find more accurate and reliable models for minimization of forecasting error. More and more annual electrical load forecasting models are submitted for raising the efficiency and accuracy of the predicted data. The conventional ones of annual power load forecasting, like regression models [1,2] and time series techniques [3,4], are rudimentary methods. Meta-heuristic optimization algorithms designed to find good solutions to optimization problems with incomplete or limited computation capacity. They have a huge ability for unscrew the complex problems node, for example, genetic algorithm (GA) [5], ant colony algorithm (ACO) [6, 7, 8 and 9], PSO [10], firefly optimizer (FFO) [11, 12 and 13] and fruit fly optimizer (FOA) [14, 15 and 16] deal with multi-objective complex optimization problems.

At recent years neural networks (NNs) are applied to power load forecasting field [17,18]. They are classified as crucial part of artificial intelligence. They emulate the human brain neurons for solving complicated problems. The primitive creation of NNs was in 1943 [19]. There are several types of NNs presented in the literature including FFN which is a type of multi-layer perceptron (MLP), Kohonen self-organizing network [20], radial base function (RBF) network [21], recurrent neural network [22], and spiking neural networks. In FFN dataflow is designed to be in one-way over the networks. Recurrent NNs enable the data to be involved all over neurons in both ways. Eventually, spiking NNs turns neurons on with spikes. Despite the different concepts of NNs, they are commonly learning (i.e. the ability to learn from expertise). NNs have the ability to imitate biological neurons and confirm themselves to a given data. The prevalent learning types are supervised [23,24] and unsupervised [25,26]. NNs need a trainer which is the method that confirms the NNs neurons and weights to act with high efficiency for the purpose of accurate results. The trainer rule is finished when the training process is done, and our net is valid to be used. So the trainer is considered as the most significant thing of any NNs. Optimization algorithms are utilized to train NNs and some of the most prevalent trainers are: GA [27], PSO [28], ACO [29] and FOA [30]. For few years grey wolf optimization algorithm (GWO) is proposed by Seyedali Mirjalili and et al [31], that imitates the natural scouting hierarchy and hunting technique of grey wolves. Four types of grey wolves like alpha, beta, delta, and omega are the scouting hierarchy. There are three essential procedures for doing hunting, alpha wolf searches prey, beta wolf encircles the prey, and delta one attacks prey. Seyedali Mirjalili and et al firstly proposed a model of training a MLP based on GWO [32], but it lacks for implementation on a forecasting problem. In this paper a modified grey wolf optimizer (MGWO) is proposed and utilized to train FFN, named MGWO-FFN. The proposed algorithm based on training FFN with MGWO, by utilizing MGWO algorithm to calculate the optimum variables of FFN based on network's performance, and after that using the yielded FFN to the LTLF problem. The work utilizes three algorithms MGWO-FFN, GWO-FFN and PSO-FFN, and deals with two cases, firstly testing the proposed algorithm by comparing it with another paper [33] that presents a study of annual electrical total consumption load forecasting for Beijing's city using a hybrid model FOA-GRNN and other algorithms. The second part is forecasting the

annual PL consumption to Egypt depending on the data gathered from the annual reports of the Egyptian ministry of electricity and energy [34]. Results of the both cases give the indication that the proposed MGWO-FFN not yet achieves more accuracy by minimizing the mean squared error MSE, but also the implementation time is shrinking compared to the other algorithms. A more accurate predicted data of Egyptian PL annual consumption is obtained and listed. The main contributions of this work are stated as follows.

- Introduce a modified version of GWO named MGWO based on feed forward net (FFN) for load forecasting problems.
- New formulation of the balancing parameter is presented for effective searching.
- Comprehensive simulations are carried out to investigate the performance on two real cases then the proposed model is affirmed by comparing other models, whereas its accuracy and validity.

The reset of this paper is organized as follows: Section 2 introduces the FFN and optimization algorithms. Section 3 presents the proposed modified grey wolf optimizer and illustrates in detail the proposed optimization algorithms included PSO, GWO and MGWO can be utilized to train FFN effectively. Section 4 presents the comparison case for forecasting of annual total electrical load consumption of Beijing and then the results of the real experimental case of long term peak load forecasting of Egypt. Section 5 concludes this paper and future work.

2 Methods and Materials

2.1 FFN architecture

FFNs are the NNs of one-way and one-directional interactions between their neurons. In this model of NNs, neurons are configured more than one layer. The first one is the input layer with four neurons that receives the data sets which are divided into training sets and testing sets each set consists of four samples (X_1 , X_2 , X_3 and X_4) as described later, whereas the last layer is named the output layer. Other layers between the input and output layers are the hidden layers. A FFN with one hidden layer is called MLP as illustrated in fig. 1. Working on FFN demands some several steps:

- a. Defining the input and output features, and this is related to the studied field.
- b. Calculating the number of hidden neurons according to the relation:

$$N_h = (N_i * N_o)^2 + 1 \quad (1)$$

Where N_h is the hidden neurons number, N_i is the number of features input of the network and N_o is the number of output layer.

c. Modeling FFN mathematically and choosing the appropriate activation functions for each layer as follows:

Inputs and bias are designed through a vector and is called input vector, then the set of weights that correlates input layer to hidden one is applied as the following relation which calculates the input of hidden neurons:

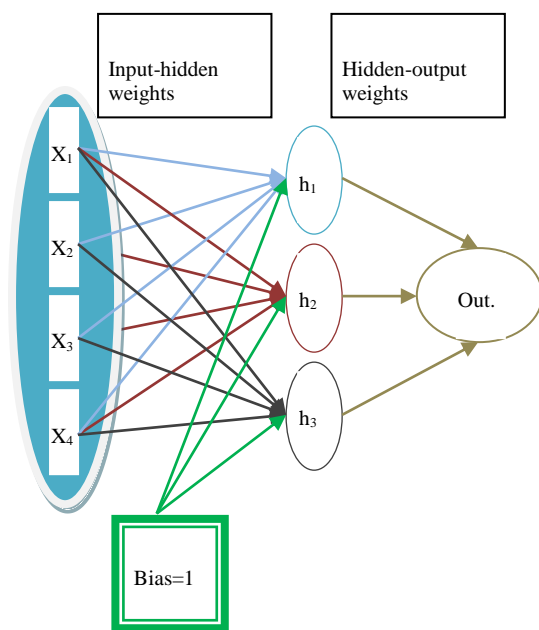


Fig.1 MLP with one hidden layer

$$h_in_j = \sum_{i=1}^n (w_{ij} \cdot X_i), \quad j = 1, 2, \dots, h \quad (2)$$

Where n is the input vector size, h is the number of hidden layer neurons, w_{ij} presents the interconnection weights between the i th node in the input layer and the j th node in the hidden layer, X is the input vector and h_in is the input value for each hidden neurons.

In order to find the output of the hidden layer node, an activation function is experimentally defined as:

$$h_out_j = \tanh(h_in_j) \quad j = 1, 2, \dots, h \quad (3)$$

Similarly the input to the last layer through another set of weights is calculated as above and is presented in the coming:

$$out_k = \sum_{j=1}^h (w_{jk} \cdot h_out_j), \quad k = 1, 2, \dots, m \quad (4)$$

Where m is the number of output layer's neurons

and w_{jk} is the interconnection weight between the hidden layer neurons and the output ones, finally the output of the last layer (prediction value) is calculated through a predetermined activation function as follows

$$y_k = \tan^{-1}(out_k) \quad (5)$$

Data were divided to two different sets training samples and testing ones and the selection of these two groups was randomized in which the training samples are around 91% of all samples and the other are the test ones. Equations (2) and (4) pointing to the importance of finding the optimal values of weights for accurate results from MLP. The next section shows different methods used to train a MLP.

2.2 Particle swarm optimizer

PSO algorithm is considered one of evolutionary computing technique referred to as swarm intelligence [10]. Such techniques mimics the flocks of birds, schools of fish and other similar social behaviors found in nature. In PSO, each flock member is considered to be a particle and they abound in the food area in a random manner, so there variables are positions, velocities and distance between them and food as the solution. The following equations illustrate the nature of how each particle updates its position according to the position of the food.

$$v_j(i) = v_j(i-1) + c_1 r_1 [P_{best,j} - x_j(i-1)] + c_2 r_2 [G_{best} - x_j(i-1)], \quad j = 1, 2, \dots, N \quad (6)$$

$$x_j(i) = x_j(i-1) + v_j(i) \quad (7)$$

Where c_1 and c_2 are the cognitive and social learning rates, respectively, and r_1, r_2 are uniformly distributed random numbers in the range 0 and 1. The parameters c_1 and c_2 denote the relative importance of the position of the particle itself to the position of the swarm. The values of c_1 and c_2 are usually assumed to be 2 so that $c_1 r_1$ and $c_2 r_2$ ensure that the particles would overfly the target about half the time.

2.3 Conventional grey wolf optimizer

Grey wolf optimizer is one of the modern swarm intelligence. Grey wolves are considered to be at the forefront of the food chain. The role of each of them can be organized in hierarchy of three levels as shown in fig. 2. GWO's hierarchy clarifies the concept of decision making for hunting. Like human groups there must be a team leader whose mission is to give orders for those under him. Then orders are delivered to the whole flock, the entire flock acknowledges are collected and then the leader make updates to decisions according to underling acknowledgement. There are three best positions (i.e., α , β and δ) to guide the whole flock for the best area where the prey exists.

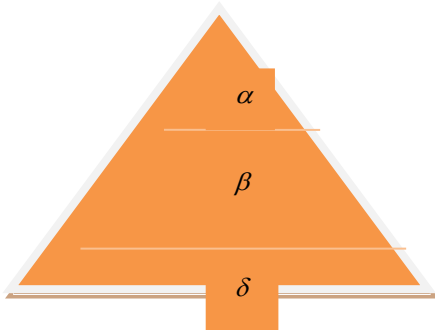


Fig.2 Hierarchy of grey wolf's order of dominance

The position of each wolf of the group is updated using encircling manner of the prey which is presented by the coming equations:

$$\vec{D} = |\vec{C} \cdot \vec{X}_P(i) - \vec{X}(i)| \quad (8)$$

$$\vec{X}(i+1) = \vec{X}_P(i) - \vec{A} \cdot \vec{D} \quad (9)$$

Where i shows the current iteration, \vec{A} and \vec{C} are coefficient vectors, \vec{X}_P is the position vector of the prey, and \vec{X} indicates the position vector of the wolf. Vectors \vec{A} and \vec{C} are calculated as:

$$\vec{A} = 2a \cdot \vec{r}_1 - a \quad (10)$$

$$\vec{C} = 2\vec{r}_2 \quad (11)$$

Parameter a denotes a linearly decreasing from [2 to 0] over all iterations as follows:

$$a = 2 - 2 * i / I_{\max} \quad (12)$$

Where I_{\max} = no. of iterations and \vec{r}_1, \vec{r}_2 have random values between 0 and 1. As can be concluded, the wolf's site X can be updated till it hits the goal or the prey by reconfiguration of \vec{A} and \vec{C} vectors.

The procedure of chasing and hunting begins when alpha, beta and delta have the best information about the prey so the hunting manner is mathematically represented by the following equations:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \quad (13)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta, \quad (14)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \vec{D}_\delta$$

$$\vec{X}(i+1) = \frac{1}{3}(\vec{X}_1 + \vec{X}_2 + \vec{X}_3) \quad (15)$$

3 The proposed MGWO-FFN algorithm

3.1 Modified GW algorithm

In this section, one takes a detailed look at a new version of GWO algorithm which is the MGWO. Exploration and profiteering are the two features that distinguish the predatory nature of GWs. Through the all iterations, firstly original GWO begins with gathering data by traveling all flock to explore the search area about the prey's position. Then data are stored in best three wolves (α , β and δ). So exploration and profiteering phases are made by adjusting the parameters a and \vec{A} . If $\|\vec{A}\| \geq 1$ the flock is in the exploration phase and for $\|\vec{A}\| < 1$ profiteering starts. For more balance between exploration and profiteering phases, a new balancing parameter a is modified as follows:

$$a = 2 - 2 * (i)^{2.6} / (I_{\max})^{2.6} \quad (16)$$

Relation (16) presents an exponential decreasing between 2 and 0 that enables the population to be more profiteers and achieve the global minimum accurately, which is different from traditional one. Fig. 3 shows the difference between the two curves of a for the basic GWO and MGWO.

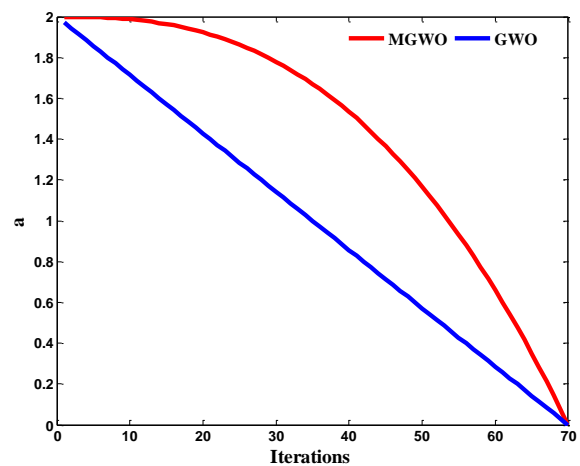


Fig.3 the difference in a shape of basic GWO and MGWO

Another modification to the candidate algorithm is implemented by proposing a new family of wolves,

which consists of the union of old β and δ families to form a new family β as follows:

$$\vec{X}_{2-new} = \frac{1}{2}(\vec{X}_2 + \vec{X}_3) \quad (17)$$

Then the final decision about the best position is taken as follows:

$$\vec{X}(i+1) = \frac{1}{2}(\vec{X}_1 + \vec{X}_{2-new}) \quad (18)$$

3.2 Training FFN based optimization algorithms

This portion clarifies how MGWO, GWO and PSO are implemented for training MLP, where all MLP parameters weights and biases as:

$$\vec{V} = \{ \vec{W}_{ih}, \vec{W}_{ho} \} \quad (19)$$

Where \vec{w}_{ih} are the set of weights which correlate the input layer to the hidden one and \vec{w}_{ho} are the correlation between the hidden layer and the output layer, and those are considered the variables of MLP. Then we must state the algorithm's objective function. As mentioned before the procedure of training MLP is to modify all weights of inputs and biases to achieve accurate predicted results by minimizing the mean squared error (MSE) of the training process by making the MLP's output meets the its actual ones (desired values) which is considered the goal of each one of the used optimization algorithms. MSE can be calculated as follows:

$$MSE = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^k (y_i - d_i)^2 \quad (20)$$

Where m is the output number, d_i^k is the actual output of the i th input unit when the k th training sample is used, and y_i^k is the predicted output of the i th input unit when the k th training sample is the input. Fig. 4 summarizes how optimization is utilized to train a MLP. The flowchart of the forecasting process is shown in fig.5

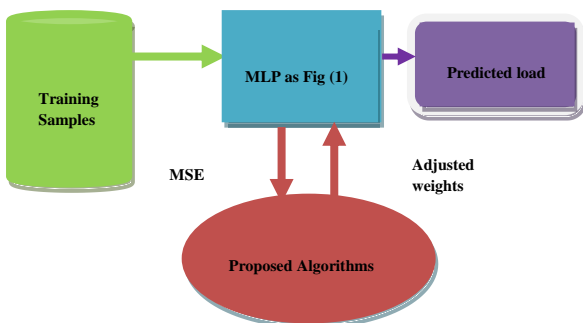


Fig.4the training process of FFN using optimization algorithms

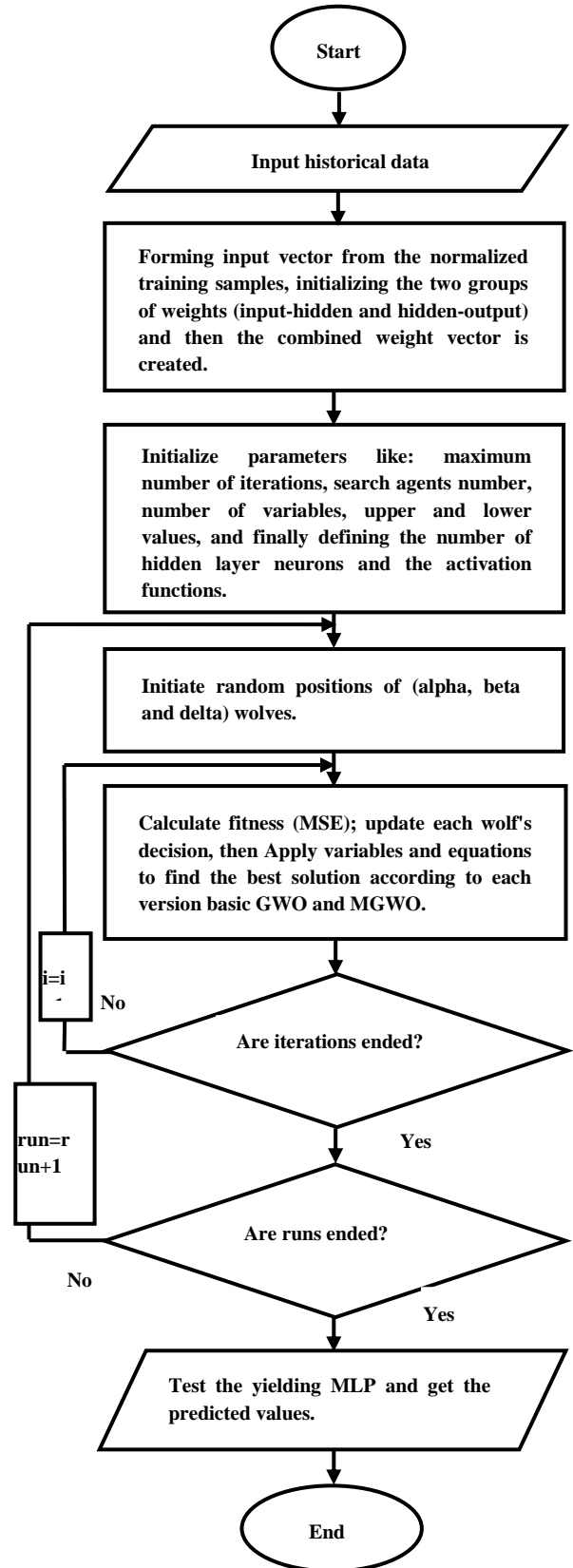


Fig.5 the flowchart

4 Simulation results

This section presents two cases of LTLF using the proposed models. Firstly, the work presents a comparison test to forecast the annual electrical total load of Beijing's city in order to prove that the proposed models outperform the others used in that paper, and this study is implemented from year 1978 to 2010 [33]. After that the real case of forecasting the Egyptian annual PL consumption with accurate results using data gathered from Egyptian electrical power system records throughout years from 1981 to 2018 [34]. Algorithms are implemented using MATLAB program 2013a with an Intel® core™ i5-7200u, 2.5GHZ CPU, 4GB RAM with windows 10 professional lap. Table 1 presents the adjusted parameters of the utilized algorithms.

Table 1: Parameters of the various algorithms used

Model	Parameters
MGWO-FFN	a decreases from 2 to 0 non-linear
GWO-FFN	a decreases from 2 to 0 linearly
PSO-FFN	$w_{max}=0.9, w_{min}=0.8, c_1=0.1, c_2=0.7$
For all algorithms	Maximum iterations=80, search agents number=70, for the test case the upper boundary=2 and the lower boundary=-3, for the real case the upper boundary=1.2 and the lower boundary=-1.7 and number of runs=50

4.1 The strategy of the forecasting process

The strategy utilized in this work can be illustrated in the following steps:

- Sampling gathered data

Historical data of consumed PL were gathered and are normalised to be in the range among -1 and 1 using the MATLAB function (mapminmax).

- Forming the input samples

The features that are used as input to the candidate models in this work are the last four annual load data ($L_{n-4}, L_{n-3}, L_{n-2}, L_{n-1}$) and the result is (L_n) i.e. to forecast the PL of year (2000) the input features required are the consumed PL of last four years before it (1996, 1997, 1998 and 1999), then to forecast the PL of year (2001) the input features are the consumed PL of years (1997, 1998, 1999 and 2000) and the used PL for year (2000) is the actual one and so on. Such technique was proposed in [33]

to forecast the annual electrical total load consumption of both Beijing and China.

- Training MLP using the proposed algorithms

The input features are classified into two parts training and testing samples, and each of them forms a ratio from all data. Each of the study cases has diverse ratios of training and testing samples. In the comparison case the training samples from year 1982 to 2005 which are 83% of the whole samples and the rest are test. The case of forecasting Egyptian PL is implemented using 82% of the whole samples through the years from 1985 to 2012 and the others are test.

- Forecasting step which is independent on the successful training process

This work is implemented using MGWO-FFN and is compared to GWO-FFN [32] and PSO-FFN.

4.2 The test case

In order to show the validity of the proposed algorithm for different cases, this paper applies the proposed algorithms included MGWO-FFN, GWO-FFN and PSO-FFN to one of the two study cases presented in [33], in which the proposed model is FOA-GRNN that was utilized to forecast the annual total electricity consumed for Beijing's city. However, GRNN's property is the ease of training for non-linear relations, the performance of it increases with optimization to recognize such patterns. Such paper compares the results of forecasted data with unoptimized-GRNN, PSO-GRNN, SAL-SSVM and OLS-LR, and FOA-GRNN outperforms other models. In this paper a comparison is presented to solve the problem case of forecasting of Beijing's city with the proposed algorithms. Results indicate supremacy of MGWO-FFN on other models because of the minimum value of MSE of value 0.7554 comparing to the MSE of FOA-GRNN which was 1.421 and less time of running whereas the FOA-GRNN algorithm took 51s for run and the proposed MGWO-FFN takes 39.5s for run. This work is done using the data presented in [33] of annual total electricity consumption of Beijing's city. The used data is normalized method using in this work, and then the testing results of the comparison algorithms are presented in table 2. Such table presents with a clear vision how MGWO-FFN outperforms the other ones by calculating MSE of the tested data, where it's MSE is 0.92 and that of the FOA-GRNN is 1.42. The MSE of the candidate algorithms is ranked to for easy arrangement of them form terms of efficiency however FOA-GRNN outperforms

MGWO-FFN in the years 2006, 2009 and 2010, but in the final MGWO-FFN achieves minimum MSE. Fig.6 presents the error analysis of the comparison case for the testing process through years from 2006 to 2010 for all models used. The figure indicates that the proposed MGWO-FFN outperforms the other models presented in that paper, not only in achieving more accurate results with more minimum MSE but also the proposed model achieves these results during few seconds comparing to the models used in that paper. From analysis curve one can see to what degree the proposed model supremacy the other models, with the improvement in performance and less running time.

Table 2: The comparison test of forecasting results of the testing process for annual total Electrical PL consumed of Beijing's city using the proposed algorithms (unit: 10^9 kWh)

Year	2006	2007	2008	2009	2010	MS E	Rank
Actual load	61.89	67.51	70.8	75.9	83.1	-	-
MGWO-FFN	62.15	67.4	70.8	77.93	82.5	0.92	(1)
GWO-FFN	61.95	65.41	72.46	77.64	82.25	2.184	(4)
PSO-FFN	62.15	66.09	73.45	77.79	82.64	2.57	(5)
FOAGRNN	61.98	67.14	72.91	77.48	83.09	1.42	(2)
GRNN	63.69	66.38	72.29	77.97	80.96	3.11	(6)
PSOGRNN	62.07	67.21	75.03	77.49	82.67	4.13	(7)
SALSSVM	62.02	67.43	73.27	78.06	82.93	2.16	(3)
OLSLR	62.14	68.13	74.61	79.43	85.07	6.27	(8)

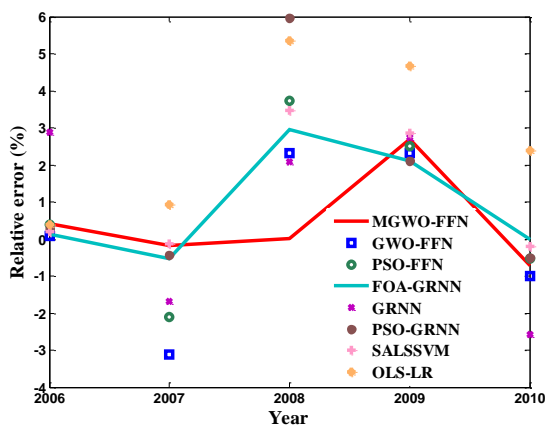


Fig.6 Error analysis curve of the testing process of Beijing's city case

4.3 Study case of Forecasting Egypt's electrical annual PL consumption using the proposed models

This part shows the study case forecasting the annual PL of Egypt using the data which are gathered from the annual reports of the Egyptian Ministry of Electricity and Energy [34] starting from year 1981 till year 2018 and presented in table 3. This study is implemented using training samples

from year 1985 to year 2012 which are 82% of the whole samples as mentioned before and the test ones are the rest of the samples from 2013 to 2018. The utilized strategy that used is the roll based strategy (i.e. roller) where our first roll is configured as input PL values of years from 1981 to 1984 and the output of it is the PL value of year 1985. The MSE of the candidate sample is estimated and stored to GWO. Then the second roll is as the same concept of the first one, so the inputs are power load data from 1982 to 1985 and the result is the predicted load of 1986, knowing that the value of the load at year 1985 used in the second roll is the actual one.

In order to show the accuracy of the proposed models fig.7 presents the fitting curve of Electric PL for the three algorithms, in which MGWO-FFN achieves minimum MSE and gives more accurate results. Such curve includes the PL data from 1985 to 2012 and the three models accuracy clears in that figure.

Table 3: Electrical PL consumed of Egypt from year 1981 to 2018 (unit: 10^9 kWh)

Year	PL	Year	PL
1981	19.036	2000	64.807
1982	21.546	2001	69.431
1983	24.63	2002	74.947
1984	26.175	2003	80.655
1985	28.664	2004	85.781
1986	31.213	2005	92.829
1987	33.42	2006	98.812
1988	35.084	2007	107.226
1989	37.415	2008	112.617
1990	39.324	2009	120.180
1991	40.793	2010	126.934
1992	42.014	2011	135.838
1993	43.610	2012	140.918
1994	46.398	2013	143.585
1995	49.534	2014	146.645
1996	52.779	2015	151.606
1997	57.106	2016	156.3
1998	57.142	2017	157.61
1999	60.863	2018	160.502

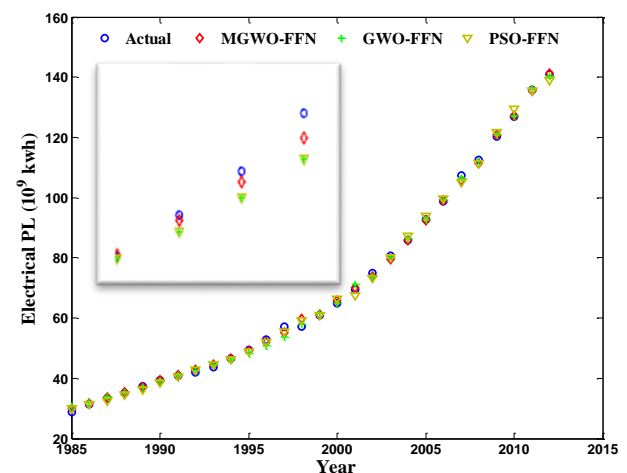


Fig.7 the PL consumption of Egypt from 1985 to 2012

MSE of the training samples during 70 iterations is presented in fig. 8 which indicates to what degree MGWO converges to minimum error of training FFN and meets the minimal MSE through first 12 iterations and increases till the end iterations as shown in the figure attached to fig.8. The rolled procedure was cycled till the load values from 1985 to 2012 are gained.

Due to the successful trained FFN using GWO, this work proposes a modified version MGOW that outperforms other methods (basic GWO and PSO) for finding the appropriate weights for FFN for minimum error between the actual and predicted ones. So that test samples are applied to the resulting net in order to measure the degree of the net accuracy.

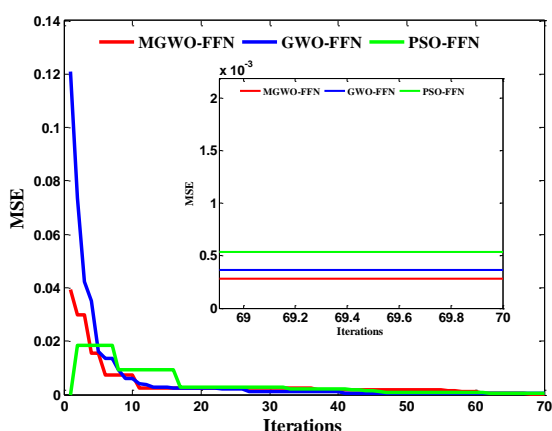


Fig.8 the iterative MSE of the proposed models

In order to prove the validity of the proposed model, data are classified into training samples and testing samples as mentioned before. So that, table 4 presents the yielding results of the testing process for years from 2013 to 2018 using the proposed models included MGWO-FFN, GWO-FFN and PSO-FFN. Error analysis is calculated according the actual load value at the testing year. Fig. 9 shows the value of the error analysis belongs to each model.

Table 4: Forecasting results of the testing process for the annual Electrical PL consumed of Egypt using the proposed algorithms (unit: 10^9 kwh)

Year	2013	2014	2015	2016	2017	2018	MSE
Actual PL	143.6	146.7	151.6	156.3	157.6	160.5	-
MGWO-FFN	143.8	146.8	151.3	155.5	156.9	160.5	0.21
GWO-FFN	144.1	146.9	152.9	155.5	158.6	160.3	0.61
PSO-FFN	143.8	148.1	152.8	154.5	156.1	160.2	1.5

The final step in the forecasting process is to use each model to predict the values of Egyptian

Electrical PL that will be consumed in the future through years from 2019 to 2021 which are presented in table 5.

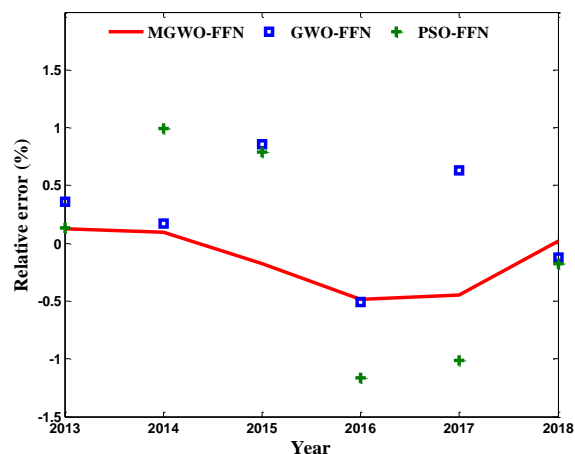


Fig.9 Error analysis curve of the testing process of Egypt case

Table 5: Forecasting results of annual Electrical PL consumed of Egypt for 2019 and 2020 (unit: 10^9 kwh)

Year	MGWO-FFN	GWO-FFN	PSO-FFN
2019	162.25	164.45	169.7
2020	165.56	168.31	172.3
2021	167.02	169.58	175.6

5 Conclusion

This paper presents a comparative study between three algorithms (PSO, GWO and MGWO) based training feed forward neural network to implement the provided two case studies, whereas the proposed model MGWO-FFN outperforms GWO-FFN and PSO-FFN and gives more accurate results for the predicted load value. MGWO-FFN converges during the 10th iteration so effectiveness and reliability is obtained through the value of mean squared error. Other advantage in this work is that the forecasting process using rolled procedure depends on the load values only and cancels other effects such that weather data, population growth number and other features. Such procedure works with data is collected from Egyptian ministry of electrical and renewable power annual records. For fair comparison this work is utilized to compare other work for the same field but that work has used FOA-GRNN to forecast the annual electrical total load forecasting of Beijing and China. The author takes one phase to the purpose of comparison (forecasting annual load consumption of Beijing's city) using the proposed three models where MGWO-FFN proves its accuracy as the reported results. The advantages of the proposed model are:

- Well trained feed forward net
- Converges to the accurate values rapidly within few times, comparing to the other stated models

The disadvantage of it may be summarized in needing some improvements to deal with large scale data.

In the future work, we investigated the proposed algorithm through using new types of neural networks for solving the electrical load forecasting with large scale power systems as well as large data set.

References:

- [1] Sorjamaa, Antti, et al. "Methodology for long-term prediction of time series." *Neuro computing* 70.16-18 (2007): 2861-2869.
- [2] H.M. Ai-Hamadi, S.A. Soliman, Long-term/mid-term electric load forecasting based on short-term correlation and annual growth, *Electric Power Systems Research* 74 (3) (2005) 353–361.
- [3] Ruijun Dong, WitoldPedrycz, A granular time series approach to long-term forecasting and trend forecasting, *Physica A: Statistical Mechanics and its Applications* 387 (13) (2008) 3253–3270.
- [4] S.Sp. Pappas, L. Ekonomou, P. Karampelas, D.C. Karamousantas, S.K. Katsikas, G.E. Chatzarakis, P.D. Skafidas, Electricity demand load forecasting of the Hellenic power system using an ARMA model, *Electric Power Systems Research* 80 (3) (2010) 256–264.
- [5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*: OUP USA, 1999.
- [6] Dorigo, Marco, and Mauro Birattari. *Ant colony optimization*. Springer US, 2010.
- [7] Rizk-Allah, Rizk M., Elsayed M. Zaki, and Ahmed Ahmed El-Sawy. "Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems." *Applied Mathematics and Computation* 224 (2013): 473-483.
- [8] Rizk-Allah, Rizk M. "Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems." *Journal of Computational Design and Engineering* 5.2 (2018): 249-273.
- [9] Rizk-Allah, Rizk M., Ragab A. El-Sehiemy, and Gai-Ge Wang. "A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution." *Applied Soft Computing* 63 (2018): 206-222.
- [10] Kennedy, R. "J. and Eberhart, Particle swarm optimization." *Proceedings of IEEE International Conference on Neural Networks IV*, pages. Vol. 1000. 1995.
- [11] El-Sawy, Ahmed Ahmed, Elsayed M. Zaki, and R. M. Rizk-Allah. "A novel hybrid ant colony optimization and firefly algorithm for solving constrained engineering design problems." *Journal of Natural Sciences and Mathematics* 266.1667 (2013): 1-44.
- [12] Rizk-Allah, R. M. "A novel multi-ant colony optimization for multi-objective resource allocation problems." *Int J Math Arch* 5.9 (2014): 183-192.
- [13] R. M. Rizk-Allah, An Improved Firefly Algorithm Based on Local Search Method for Solving Global Optimization Problems, *International Journal of Management and Fuzzy Systems* 2016; 2(6): 51-57.
- [14] Allah, Rizk M. Rizk. "Hybridization of fruit fly optimization algorithm and firefly algorithm for solving nonlinear programming problems." *International Journal of Swarm Intelligence and Evolutionary Computation* 5.2 (2016): 1000134.
- [15] Rizk-Allah, Rizk M., et al. "A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor." *The Journal of Supercomputing* 73.3 (2017): 1235-1256.
- [16] Rizk-Allah, Rizk M., Aboul Ella Hassanien, and Siddhartha Bhattacharyya. "Chaotic crow search algorithm for fractional optimization problems." *Applied Soft Computing* 71 (2018): 1161-1175.
- [17] Feng, Li, and Ziyang Liu. "Genetic algorithms and rough fuzzy neural network-based hybrid approach for short-term load forecasting." 2006 IEEE Power Engineering Society General Meeting. IEEE, 2006.
- [18] Swaroop, R., and H. A. Abdulqader. "Load forecasting for power system planning and operation using artificial neural network At Al Batinah Region Oman." *Journal of Engineering Science and Technology* 7.4 (2012): 498-504.
- [19] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5.4 (1943): 115-133.
- [20] Kohonen, Teuvo. "The self-organizing map." *Proceedings of the IEEE* 78.9 (1990): 1464-1480.
- [21] Park, Jooyoung, and Irwin W. Sandberg. "Approximation and radial-basis-function

networks." *Neural computation* 5.2 (1993): 305-316.

- [22] Dorffner, Georg. "Neural networks for time series processing." *Neural network world*. 1996.
- [23] Reed, Russell, and Robert J. MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.
- [24] Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006.
- [25] Hinton, Geoffrey E., Terrence Joseph Sejnowski, and Tomaso A. Poggio, eds. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.
- [26] Wang, DeLiang. "Unsupervised learning: foundations of neural computation." *AI Magazine* 22.2 (2001): 101-101.
- [27] Montana, David J., and Lawrence Davis. "Training Feedforward Neural Networks Using Genetic Algorithms." *IJCAI*. Vol. 89. 1989.
- [28] Mendes, Rui, et al. "Particle swarms for feedforward neural network training." *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. Vol. 2. IEEE, 2002.
- [29] Blum, Christian, and Krzysztof Socha. "Training feed-forward neural networks with ant colony optimization: An application to pattern classification." *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*. IEEE, 2005.
- [30] Xu, Z. H., et al. "A forecast of export trades based on the FOA-RBF neural network." *Math. Pract. Theor* 42 (2012): 16-21.
- [31] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in engineering software* 69 (2014): 46-61.
- [32] Mirjalili, Seyedali. "How effective is the Grey Wolf optimizer in training multi-layer perceptrons." *Applied Intelligence* 43.1 (2015): 150-161.
- [33] Li, Hong-Ze, et al. "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm." *Knowledge-Based Systems* 37 (2013): 378-387.
- [34] http://www.moe.gov.eg/test_new/report.aspx

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US