Neural Network Training Model for Weather Forecasting Using Fireworks Algorithm

Saktaya Suksri
Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: 58605077@kmitl.ac.th

Warangkhana Kimpan, *Member*, *IEEE*
Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: warangkhana.ki@kmitl.ac.th

Abstract—**Weather forecasting is the application of science and technology in order to predict the weather conditions. It is important for agricultural and industrial sectors. Models of Artificial Neural Networks with supervised learning paradigm are suitable for weather forecasting in complexity atmosphere. Training algorithm is required for providing weight and bias values to the model. This research proposed a weather forecasting method using Artificial Neural Networks trained by Fireworks Algorithm. Fireworks Algorithm is a recently developed Swarm Intelligence Algorithm for optimization. The main objective of the method is to predict daily mean temperature based on various measured parameters gained from the Meteorological Station, located in Bangkok. The experimental results indicate that the proposed method is advantageous for weather forecasting.**

Keywords—weather forecasting; fireworks algorithm; artificial neural network; numerical weather prediction; swarm intelligence

I. Introduction

Weather forecasting plays an important role in our lives. It is used to predict or estimate natural phenomena that occur during a certain period of time, for example, predicting the average temperature of the day and the following days, sunny days or rainy days including sea waves and earthquakes. The mentioned phenomena can affect our living both everyday life and careers. As nature is difficult to anticipate in the meanwhile human need to know what is going to happen in the near future so they can manipulate their routine determination in order to make their economy grow via all sorts of projects, activities or businesses, including agriculture and industries. To be able to run their activities or businesses and avoid risks, they certainly need the weather forecast. However, the weather forecast is not an easy task without appropriate data collection. Thus, weather forecast requires a long-term data collection and statistics. Researchers have used the previous data accumulated through several years to be able to analyze the relationships of each phenomenon. Eventually, computers have come to play significant roles in data collection. At the present, there is a computer model to assist in forecasting called Numerical Weather Prediction (NWP). NWP is an objective forecast that is predicting the state of the atmosphere by applying dynamics, thermodynamics, and statistics using numerical model for calculation of the change of variables related to weather conditions by nonlinear equation.

Artificial Neural Networks (ANNs)[1] is a complex mathematics model inspired by the model of neurons in the human brain that can recognize and learn by imitating the human brain. It has structural models to solve a variety of formats. Each model has a different way out, as well as the pros and cons for each. One popular type used in prediction is the Supervised Learning, which requires sophisticated algorithms in ANNs training to recognize the early state to predict the outcome. The most popular of training algorithm is the Backpropagation Algorithm[2]. ANNs can solve weather forecasting[3] because of the abilities to find relationships of variables within data series. The knowledge of ANNs acquired from the training algorithm determines the proper value. The algorithm can be trained using historical data as input data in ANNs models for processing and validation. Then, the process to find the optimal value is repeated. When the training was successful, the predicting output value of datasets may come from the same source or any similar source.

Fireworks Algorithm (FWA) is a novel swarm intelligence algorithm introduced by Ying Tan[4]. FWA is one method of collaborated intelligence groups, which is a technique that simulates the behavior of fireworks in the night sky. The algorithm is generally used for solving optimization by searching for the optimum value inside the solution space of specific problems.

This research proposed the methodology of training ANNs by using the FWA to determine the weights and biases that interconnect in the network. The main idea of this research is to present effectiveness of the ANNs trained by using FWA for solving weather forecast. The initial purpose is assumed that the model has fast convergence toward the answer. The proposed model has high accuracy and appropriate complexity; moreover, the model produces acceptable prediction with new datasets in a feasible time. This paper is organized as the followings: Section II presents the methods of adapting ANNs for solving weather forecast. Section III Fireworks Algorithm is explained. Section IV illustrates the proposed method of developing training ANNs using FWA for weather forecasting. Section V presents

978-1-5090-4420-7/16/$31.00 ©2016 IEEE

experimental methods and results. Finally, Section VI is conclusions and future works.

II. Artificial Neural Networks and Related Works

A. *Artificial Neural Networks for Weather Forecasting*

ANNs have been applied to solve problems in both classification and regression of weather states. In the case of using ANNs for weather forecasting[5], the networks are created to predict acceptable criteria of the output value. The networks are built from changing or adjusting the knowledge of the system in predicting the output value close to the target value. Analysis of problem is needed to create the model of ANNs. The characteristics and quantity of data directly affect time processing, performance training, and the system failure.

1) *Artificial Neural Networks Architecture*

Prior to training, the problem domain must be analyzed, and then the components of the ANNs are determined. Neurons in the network are organized in layers; each layer provides a particular level of processing. The number of neurons in input layer depends on the number of input data attributes while the number of neurons in output layer depends on the determination of predictive output. Generally, regression problems have a single neuron for predicting the numerical output. The number of neurons in hidden layer is determined by research and experiment. In addition, activation functions which are important factors in making the networks to process as nonlinear, are selected by considering the appropriateness of the datasets. Additionally, weight values and bias values are other important factors. Weights are the connected neurons processing values. The number of weights can be calculated as:

$$\text{numWeights} = (\text{numInput} \cdot \text{numHidden}) + (\text{numHidden} \cdot \text{numOutput}) \quad (1)$$

Biases are important parameters which can help ANNs produce nonlinear system by adding extra signal to the activation functions. The number of biases is defined as:

$$\text{numBiases} = \text{numHidden} + \text{numOutput} \quad (2)$$

After defining the components, it is necessary to determine the pattern of ANNs processing. All ANN architectures are designed for a specific task. This paper uses Multilayer Feedforward for the network structure with only one hidden layer between input and output layer. In this category, neurons are split into multiple layers; each layer consists of a parallel layout of neurons. When data is entered into input layer of the model, it is processed and then entered into the next layer of the network for the next processing.

2) *Artificial Neural Networks Learning*

In this research, learning of ANNs is a supervised learning paradigm. The aim of the learning is knowledge, weights, and biases. The learning will be tested to measure the efficiency by comparing the predicted output of network processing with the real output. This is called validation. Basically, the learning process is modeled as below:

$$y = O(w,e)+Err \quad (3)$$

where y is the predicted outputs that network produced
O is the activation function
w is the matrix of weights and biases
e is the matrix of input and real output for training
Err is the error between real and predicted outputs

Training ANNs is a process of minimizing the error value by optimization. The goal of training is to produce the output with zero error, or as close to zero as possible. At this training stage, there are various algorithms to perform training of ANNs. Ideally, the learning process needs training algorithm then the algorithm calculates weights and biases, and the value is updated continuously. The training process is terminated by acceptation criteria.

B. *Related Works*

In 2007, M. Hayati and Z. Mohebi[6] and B. A. Smith, R. W. McClendon and G. Hoogenboom[7] aimed to develop predictive models of ANNs for reducing average error. The results showed low error value of the prediction, and how the model was used for forecasting the short term. In 2008, M. Nasseri, K. Asghari and M. J. Abedini[8] proposed a Feedforward multilayer perceptron for predicting rain. The training was processed by using a couple of Backpropagation Algorithm with Genetic Algorithm (GA). The model represented outperforms traditional result. Later in 2012, K. Abhishek, A. Kumar, R. Ranjan and S. Kumar[9] proposed the ANNs model for predicting the average rainfall of each month. The model used 400 records of Udupi, Karnataka, and all records were divided into 70% for training and 30% for testing. The model was implemented by using the mean and standard deviation for normalization and Backpropagation Algorithm for training networks with three layers. After processing, the predicted results were compared with the expected results, and found a high similarity. Thus, this research could prove that the model can predict accurate results. C. J. Devi, B. S. P. Reddy, K. V. Kumar, B. M. Reddy and N. Raja[10] aimed to predict time series data by using Min-Max method for normalization. The results of experiment produced proper accuracy. In 2015, A. G. Salman, B. Kanigoro and Y. Heryadi[11] proposed deep learning techniques for weather forecasting. The research compared the prediction performance of the Recurrence Neural Network, Conditional Restricted Boltzmann Machine, and Convolutional Network. These models were tested using real weather dataset and using Frobenius norm for evaluating the accuracy of each model. The results showed that the Recurrence Neural Network can be applied in predicting rainfall with an adequate accuracy level. A. Grover, A. Kapoor and E. Horvitz[12] proposed a deep hybrid model for weather forecasting by combining discriminatively trained predictive models with a deep neural network. The experiments used real-world meteorological data for evaluating the methods. The results indicated that the model can provide better results than the National Oceanic and Atmospheric Administration benchmarks.

III. Fireworks Algorithm

Fireworks Algorithm (FWA) is Swarm Intelligence algorithm inspired by a group working social behavior. This algorithm is mainly used for optimization. It simulates the explosion of fireworks at night. The night sky refers to a solution space and the firework explosion refers to a search for answers. The main process starts with generating some sample values by randomization and then evaluates a suitable result. After that, the selection process for the next generation is executed. The parameters are adjusted before calculating for the next cycle and repeated the cycle until the right answer for the solutions is met.

An explosion of fireworks occurs when fireworks are shot up into the sky, and they move to the appropriate position and explode into shining lights in various shapes, depending on the designs of the fireworks. Each firework has a specific explosion characteristic, for example, a narrow explosion with a lot of sparks, a wide explosion with fewer sparks, or a wide explosion with so many sparks, and so on. Thus, each firework has a unique explosion and different position to set off.

In this research, FWA is used as an algorithm for training ANNs and determining the proper weights and biases. The proper values are chosen from the processing of the error values between the predicted results and the actual results. This algorithm has the characteristics of a repeating cycle until a suitable result is achieved. The work is divided into four main parts as described in the followings.

1) Explosive Operator

Sparks are built around the explosive fireworks inside the amplitude of fireworks. The breadth and the number of the sparks will be controlled at this stage. During the initial of FWA, N number of fireworks is created. Each firework establishes various amounts of sparks and explosion amplitude depending on the explosive operator. The blast in this step is an important stage of the algorithm. This process consists of three sub-processes as followed:

a) Explosion Strength

Explosion strength is the main sub-process of the explosion. It assigns the number of sparks and calculates the amplitude for each firework. The good explosion of fireworks causes better fitness and yields the optimal value. It has a lot of sparks with a narrow range that means the frequency of sparks is high in the proper amplitude. On the other hand, the bad explosion has a wide range of explosion with fewer sparks. It causes the worst fitness. The advantage of the firework explosion with the bad explosion is used for preventing the premature convergence that is the quick convergence to the answer of the algorithm. At this stage of the explosion strength, the number of sparks is defined as shown below:

$$S_i = m * \frac{Y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^{N} (Y_{max} - f(x_i)) + \varepsilon} \quad (4)$$

where S_i is the number of sparks for each firework
m is a constant for total number of sparks
Y_{max} is the fitness value of the worst individual
$f(x_i)$ represents the fitness for an individual x_i
ε is minimum value used to prevent the denominator from becoming zero

b) Explosion Amplitude

Explosion amplitude is used in controlling the size of the explosion. In the typical optimization method, the positions surround the solution provide the best fitness value. Therefore this process gradually reduces the width of the good explosion with the best fitness value to converge a better value. The amplitude of each firework is calculated as:

$$A_i = \widehat{A} * \frac{f(x_i) - Y_{min} + \varepsilon}{\sum_{i=1}^{N} (f(x_i) - Y_{min}) + \varepsilon} \quad (5)$$

where A_i denotes the amplitude of each individual
$\widehat{A}$ is a constant as the sum of all amplitudes
Y_{min} is the fitness value of the best individual
$f(x_i)$ and ε are the same meaning as in (4)

c) Displacement Operation

Displacement takes place after calculating the explosion amplitude. The displacement value is configured by a uniform random number within the intervals of each cycle of the explosion amplitude.

2) Mutation Operation

Mutation operation is used to improve the diversity of the population by Gaussian mutation in order to produce sparks. Only one firework is selected from the current population. Then the Gaussian mutation is applied and taken into random dimensions. The position of a new spark created by Gaussian mutation is in between current location and the origin.

3) Mapping Rule

When the explosive position of the firework near the border of the solution space happened, the width of the explosion could build up useless spark outside of the space. Therefore, it must be adapted to be within the solution space.

4) Selection Strategy

Selection strategy is executed for selecting the next generation. The best spark is stored. Meanwhile, other sparks are considered for the selection by using Euclidean distance as shown below:

$$R(x_i) = \sum_{j=1}^{K} d(x_i, x_j) \quad (6)$$

where $R(x_i)$ represents the sum of distances between individual x_i and all the other individuals
K is the set of combining both the sparks generated by explosion operator and mutation operator

The sparks with the distance from the others tend to have a higher chance of being picked for the next generation.

IV. METHODOLOGY

To forecast the weather, time series data of the weather in the past, which are continuously collected at regular intervals, are stored. Then, the data are entered through the data preprocessing process. The process is used for preparing the data to be ready for the ANNs training by FWA in order to predict the numerical output of the temperature.

A. Data Preparation

1) Data Selection

ANNs need sets of data which are relevant to the problem issue for learning. Thus, the data selection process is important. Each attribute or feature in datasets must be interrelated. The typical attributes for the weather prediction are temperature, wind speed, humidity, sun energy, sunny hour, and so on. Therefore, the selected datasets require studies of the data relationships and data resources must be reliable. The most reliable datasets used for the weather forecast are taken from the Department of Meteorology, e.g., weather stations, satellites, radar stations and the like. Generally, the datasets have multiple frequencies depending on the options used in solving the problems, e.g., hourly, daily, weekly, and monthly. Some variables are the summed up data, such as the average of the temperature value, and the minimum and maximum of the temperature value.

When the data are selected, the parameters must be chosen to define the input and output of the model. The output is what to predict. The format of each data row for model learning is required as shown below:

$$\mathrm{Data_i} = \{\mathrm{Input}_1, \mathrm{Input}_2, \mathrm{Input}_n, \mathrm{Output}_1, \mathrm{Output}_n\} \quad (7)$$

2) Data Filtering

Data filtering is a procedure for eliminating the useless data, such as the data with fault, insignificance, void, repetition, irrelevance, sensitiveness or out of bound. The useless data should be deleted, or replaced. Data filtering process can make the ANNs learn well; moreover, it can also make the results more efficient in prediction.

3) Data Normalization

The normalization is a process for scaling data to be in the same range. Basically, the data are scaled into the range between 0 and 1 or -1 and 1 in order to make the learning process more effective. This study uses Min-Max Normalization for scaling data into the range of 0 to 1 as shown below:

$$x_{norm} = \frac{(X - X_{min})}{X_{max} - X_{min}} \quad (8)$$

where X is a data value
X_{min} is the minimum value
X_{max} is the maximum value
x_{norm} is a value after normalization

B. Training Artificial Neural Networks Using Fireworks Algorithm

1) Model Implementation

This model was developed using C# from the theory of Feedforward neural network and FWA which was adapted for finding the optimal value of weights and biases. This model requires the input data to be learned, and executes the training process to determine the weights and biases of ANNs in predicting the results. The components of ANNs are defined as the followings: the number of input neurons equals to the number of features of the input datasets, the number of hidden neurons is determined by the experiments, the number of output neurons equals to 1, the activation function uses Hyperbolic tangent for scaling outputs from -1 to 1, the number of weights is referred in (1), and the number of biases is referred in (2). The weights and biases value are produced as shown below.

Algorithm : Pseudocode of training ANNs using FWA

```
Read weather datasets as input
Do filtering process
Do normalization process
Divide data into Training and Testing set
Initialize ANNs parameters, FWA parameters and maximum
number of training cycles
Generate random fireworks for the first generation
For i=1 to maximum number of fireworks do
    Calculate error;
End for
Initialize the best position
For j=1 to maximum number of training cycles do
    For i=1 to maximum number of fireworks do
        Calculate amplitude; Calculate number of sparks;
        Generate regular sparks;
    End for
    Generate Gaussian sparks;
    Evaluate sparks;
    Select number of sparks for next firework;
    Generate new number of fireworks;
End for
Return the best weights and biases value
```

2) Performance Measurement

a) Validation

Validation process is required for ANNs optimization and it reduces over-fitting. The over-fitting is likelihood of over learning, which makes a predicting model effective only for the datasets used for training, but it makes a predicting model less effective for unknown datasets. The cross-validation is used for assessing the quality of the model by dividing the datasets into training set and testing set. Moreover, this process is used to test the performance of models with different setting parameters. The processing results can be used to choose the best set of parameters. This study used k-fold cross-validation; the data were divided into k sets from which one set was used for testing and k-1 set for testing then repeated k times. Then k results will be averaged.

b) Evaluation Process

The evaluation process is a method to evaluate error values. It is executed to measure the performance. The output that is close to zero is the best output. The Root Mean Squared Error (RMSE) method was used in this study as shown below:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \times \sum_{j=1}^{N} (F_j - Y_j)^2} \quad (9)$$

where N is the number of testing data, F is the predicted value, and Y is the real value.

V. Experimental Setup and Results

A. Experimental Setup

1) Data Exploration

The datasets used in this study are daily local weather stored in the past. The datasets stored from 1 January 2012 to 31 December 2015 were taken from Chaloemprakiet Meteorological Station, located in Bangkok. Each feature has various samplings; the details are shown in Table I. The whole datasets for experiments had been daily averaged. Then, the filtering process was executed from the acquired raw data. In some cases, the data were incomplete, null or faulty storage. Thus, those data rows were deleted even if the values of attributes exist or non-exist. Finally, Min-Max normalization was executed to prepare the data for the training process.

2) Experiment Method and Parameter Setting

In the experiments, the FWA parameters for training are presented in Table II. The details of the adjusting parameters in the learning model are shown in Table III. The validation process is required. The process used 10-fold cross-validation and was divided into 9 folds for training dataset and 1 fold for testing dataset alternately repeated. In terms of training cycles, when the training reached the maximum number of the training cycles, the learning process is stopped. After that, the evaluation process for the model performance estimation was executed.

B. Experiment Result and Discussion

In the experiments, predictive models with tuning parameters were divided into four experiments as the followings: Experiment I: using local weather dataset of a one-year period from January 1, 2012 to December 31, 2012. The results are shown in Fig. 1(a). Experiment II: using local weather dataset of a one-year period from January 1, 2013 to December 31, 2013. The results are shown in Fig. 1(b). Experiment III: using local weather dataset of a two-year period from January 1, 2012 to December 31, 2013. The results are shown in Fig. 1(c). Experiment IV: using local weather dataset of a three-year period from January 1, 2012 to December 31, 2014. The results are shown in Fig. 1(d). The experiments results were concluded as followed.

1) The Influence of Number of Hidden Neurons

According to Fig. 1, when comparing each number of hidden neurons in all cases, it can be concluded that 1 and 3 hidden neurons are sufficient for the use of predicting the average temperature. The result of Experiment I and IV in Table IV indicated number of hidden neurons which equals to only 1 can produce the minimum error. However, the increment of hidden neurons has a better probability to find a better value as showed in Experiment II and III. The increment of hidden neurons makes processing more complex which directly affects the result of accuracy and time processing for the prediction.

2) The Influence of Number of Fireworks

In searching for the answer, when the Number of fireworks or amount of population, were equal to 5 and 7, the low level of error values was found in comparing to other parameters. In Fig. 1, a few number of fireworks can lead to good results because the FWA is capable of great exploitation in each population that has the higher chance of getting good results. However, the increment of this parameter has a better probability to find a better value as shown in Table IV. The number of fireworks which equals to 13 produced minimum error in Experiment II and III, meanwhile, the incremental population means the increment of the searching in the solution space.

3) The Influence of Number of Training Cycles

The experimental results in Fig. 1 and Table IV showed the maximum number of training cycles which equals to 500 and 1,000. This can make the RMSE close to zero. The FWA has a fast convergent behavior to the answer. Therefore, this training algorithm can yield accuracy to the prediction and also speed up the training process. Both lead to the reduction of the number of training cycles. So in this parameter, it is not necessary to use the high number of training cycles. Meanwhile, if the problems required high accuracy, the increment of training cycles may bring higher accuracy as shown in Experiment II of Table IV but it must be traded off with the increment of time processing.

4) The Influence of Number of Training Dataset

The summarize results of four experiments in Table IV showed minimum of error values. The Experiment III indicated the lowest RMSE value occurred when training by using local weather dataset of a two-year period. Though the dataset used in this research had been continuously collected in the Bangkok area for several years, each attribute of each dataset had similar values. This indicates that increasing the number of local weather dataset of a three-year period in

TABLE I. Training and Testing Dataset Attributes

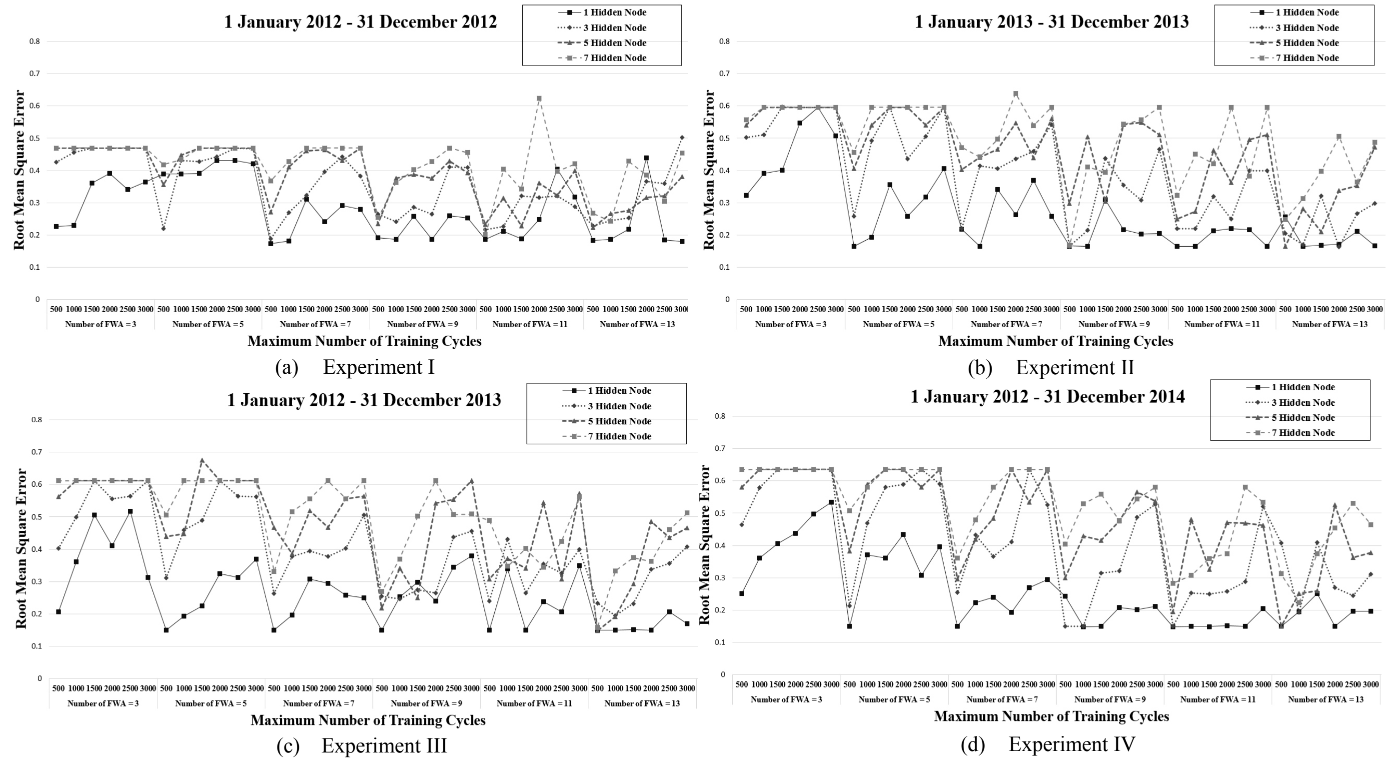
Feature	Unit	Description
Mean humidity	%	Average of hourly measurements
Precipitation	Mm	Accumulation of daily rain
Pressure	mbar	Average of 3 hour measurements
Insolation	h	Count of hours receiving sun radiation daily
Mean wind speed	km/h	Average of 3 hour measurements
Max wind speed	km/h	Accumulation of daily maximum wind speed
Wind direction	°	Average of 3 hour measurements
Mean temperature	°C	Average of hourly measurements

TABLE II. Parameters Setting of Fireworks Algorithm

Symbol	Parameter	Value
n	Number of fireworks	3 - 13
m	Total number of regular sparks	n*10
mHat	Number of Gaussian sparks	5
a	Minimum of spark in each firework	0.04
b	Maximum of spark in each firework	0.8
A	Maximum of explosion amplitude	40
minX	Minimum number for control position	-10
maxX	Maximum number for control position	10

TABLE III. Parameters Tuning

Parameter	Range of Value	Increment
Number of hidden neurons	1-7	2
Number of fireworks	3-13	2
Number of training cycles	500-3000	500



(a) Experiment I

(b) Experiment II

(c) Experiment III

(d) Experiment IV

Figure 1. Root mean square error of Experiment I to Experiment IV

Experiment IV for training to predict the average temperature using FWA as shown in Fig. 1(d) does not directly affect the accuracy. To initially sum up, there is no need to use a lot of data for the training, if the dataset has several similar parameters. This can reduce time for the training process.

5) The Influence of Predicting Unseen Dataset

The trained model was used to predict the mean temperature of an unseen dataset that comes from the same source from January 1, 2015 to December 31, 2015. The trained model has parameters setting which are taken from the random sampling among the suitable ranges as mentioned above. The parameters setting are as followed: the number of hidden neurons is equal to 1, the number of fireworks is equal to 7, the number of training cycles is equal to 1,000, and the training set is a one-year period from January 1, 2012 to December 31, 2012. After the processing; the predicted results of training set and testing set were compared with the actual results, and a high similarity was found as shown in Fig. 2 and Fig. 3. Thus, this research could prove that the model can predict accurate results. The prediction in the training set has Accuracy = 81.48%. The prediction in the testing set has Accuracy = 73.79%.

TABLE IV. THE BEST SET OF PARAMETERS

Experiment	Parameter			
	Minimum error	**Number of firework**	**Number of hidden neuron**	**Number of training cycle**
I	0.1727	7	1	500
II	0.1634	13	3	2000
III	0.1476	13	5	500
IV	0.1485	11	1	500

VI. CONCLUSIONS AND FUTURE WORKS

This research proposed training ANNs using FWA in creating a predictive model for predicting mean temperature. It can be seen that the performance of the model produces acceptable results, and the optimized model can predict the unseen dataset with the similar values. The FWA has fast convergence as well as reduces the training cycles, and it has various parameters for adjusting to find an optimal value of the specific problems.

Future works may include the comparison of this studied with other prediction algorithms or other methods. More hidden layer and tuning the other parameters can also be studied. Apart from that, we wish to integrate proposed model with other training techniques for weather forecasting.

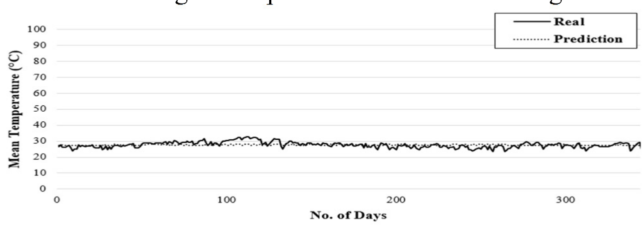


Figure 2. Real and predicted values of mean temperature in training set

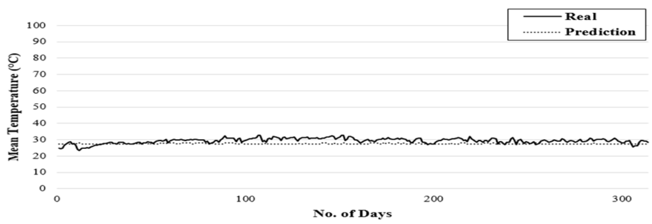


Figure 3. Real and predicted values of mean temperature in testing set

References

[1] R. J. Schalkoff, Artificial neural networks, New York: McGraw-Hill, 1997.

[2] R. Rojas, Neural Networks: A Systematic Introduction, Berlin: Springer-Verlag, 1996.

[3] D. Fletcher and E. Goss, "Forecasting with neural networks: An application using bankruptcy data," Information & Management, vol. 24, no. 3, pp. 159-167, 1993.

[4] Y. Tan and Y. Zhu, "Fireworks Algorithm for Optimization," First International Conference in Advances in Swarm Intelligence, (ICSI 2010), Part I, LNCS 6145, pp. 355-364, 2010.

[5] A. Culclasure, "Using Neural Networks to Provide Local Weather," Electronic Theses & Dissertations, 2013.

[6] M. Hayati and Z. Mohebi, "Application of Artificial Neural Networks for Temperature Forecasting," World Academy of Science: Engineering & Technology, vol. 22, pp. 275, 2007.

[7] B. A. Smith, R. W. McClendon and G. Hoogenboom, "Improving air temperature prediction with Artificial Neural Networks," International Journal of Computational Intelligence, vol. 3, no.3, pp. 179-186, 2007.

[8] M. Nasseri, K. Asghari and M. J. Abedini, "Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network," Expert Systems with Applications, vol. 35, no. 3, pp. 1415-1421, 2008.

[9] K. Abhishek, A. Kumar, R. Ranjan and S. Kumar, "A rainfall prediction model using artificial neural network," IEEE Control and System Graduate Research Colloquium (ICSGRC), pp. 82-87, 2012.

[10] C. J. Devi, B. S. P. Reddy, K. V. Kumar, B. M. Reddy and N. Raja, "ANN Approach for Weather Prediction using Back Propagation," International Journal of Engineering Trends and Technology, vol. 3, no. 1, pp. 19-23, 2012.

[11] A. G. Salman, B. Kanigoro and Y. Heryadi, "Weather forecasting using deep learning techniques," 7th International Conference on Advanced Computer Science and Information Systems (ICACSIS 2015), pp. 281-285, Indonesia, 2015.

[12] A. Grover, A. Kapoor and E. Horvitz, "A Deep Hybrid Model for Weather Forecasting," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2015), Australia, pp. 379-386, 2015.