NONLINEAR AUTOREGRESSIVE NEURAL NETWORK FOR ESTIMATION SOIL TEMPERATURE: A COMPARISON OF DIFFERENT OPTIMIZATION NEURAL NETWORK ALGORITHMS

Omaima N. A. AL-Allaf $^{\#1}$ **and** **Shahlla A. AbdAlKader** ^{*2}
*Dept. of Computer Information Systems, Faculty of Sciences and Information Technology
AL-Zaytoonah Private University of Jordan
P.O. Box 130, Amman (11733), Jordan
omaimaalallaf@alZaytoonah.edu.jo

#Dept. of Computer Systems, Foundation of Technical Education, Technical Institute/Mosul,
Ministry of Higher Education, Mosul-Iraq
shahla_ak71@yahoo.com

ABSTRACT

The objective of this paper is to develop an Artificial Neural Network (ANN) model to estimate soil temperature for any day. We used air average temperature, sunshine, radiation and soil temperature for meteorological data between years [1980 and 1984] at Nineveh/Iraq Meteorological Station.

In this research, three ANN models with their associated training algorithms (Backpropagation neural network (BPNN), Cascade-Forward and Nonlinear Autoregressive (NARX)) were used for estimating soil temperatures at different depths of 5, 10, 20, 50 and 100cm within the time 9, 12 and 15 respectively.

The performance of the three models with their training algorithms were compared with the measured soil temperature values to identify the best fit ANN model for soil temperature forecasting. The results showed that the NARX model is the best model. Finally, a comparison between five optimization ANN training algorithms was adopted to train NARX ANN model to identify best fit optimization algorithm for forecasting soil temperature with best results. From comparisons, TrainLM is the best optimization algorithm for training NARX model.

Keywords: Artificial Neural Networks, BackPropagation Algorithm, Cascade-Forward Algorithm, Time Series Algorithm, Soil Temperature

1 INTRODUCTION

Artificial Neural Networks (ANNs) were developed to emulate the non-linear functions of human natural neural networks [1]. From equations which require input/output relationships, ANNs are trained by sample data to build the input/output vector maps in an implicit way. So, ANNs can solve highly non-linear problems without the need to define the relationship between inputs and outputs [2].

The backpropagation neural network (BPNN) is a multi-layer feed forward ANNs which use backpropagation algorithm (BP) for training and it is the most popular architecture. Methods using standard BP perform gradient descent only in the weight space of a network with fixed topology. In general, BPNN is useful only when the network architecture is chosen correctly. Too small network cannot learn the problem well, but too large size will lead to over fitting and poor generalization performance [3].

Soil temperature is an important meteorological parameter for ground source heat pump applications, solar energy applications (heating and cooling of buildings), frost prediction, and other applications [4], [5].

Soil temperature determines the type and rate of different physical and chemical reactions in the soil. It affects diffusion of nutrients in soil and their uptake by plants. It influences the rate of organic matter decomposition, which in turn affects soil structure and water movement in the soil. Plant growth is more rapid when the soil temperature increases up to the optimum level. The soil temperature also affects the functional activities of plant roots such as absorption and translocation of water. Whereas, crop species differ in their response to soil temperature and each species has its optimum range of temperature for maximum growth [6].

Also, the soil temperature is an important parameter that directly affects the growth of plants and biological and physical processes occurring in the soil [7]. Finally, soil surface temperature is an important factor for calculating the thermal performance of buildings in direct contact with the soil and for predicting the efficiency of earth-to air heat exchangers [4].

From the above, many literature researches focused on determining the soil temperature. Yang, et. al. 1997 [2] developed ANN model to simulate soil temperature at different depths by considering available meteorological parameters. They used five years of meteorological data which measured at a weather station at central experimental farm in Ottawa, Ontario, Canada. The model inputs consisted of daily rainfall, evaporation and the day of the year. The model outputs were daily soil temperatures at the depths of 100, 500 and 1500 mm. Their estimated values were found to be close to the measured values. Paul et al., 2004 [8] stated that daily fluctuations in soil temperature influence biological and chemical processes in the soil. Yilmaz et al., 2009 [5] stated that determination of ground temperature at different depths is very important for agricultural and ground source heat pump applications and for the calculation of heat losses from the parts of buildings that are buried in the ground.

For these purposes, we require accurate soil temperature measurements. Soil temperature depends on a variety of environmental factors including: meteorological conditions such as surface global radiation and air temperature; soil physical parameters such as albino of surface; water content and texture; and finally topographical variables such as elevation, slope and aspect [7], [8]. Therefore, prediction of soil temperature is difficult, especially near the ground surface where the variations of the soil temperature are high [4].

In recent years, several studies were concerned on determination of soil temperatures using analytical models and experimental methods [6], [4], [8]. At the same time ANNs models have been used for soil temperature forecasting and estimation.

Hayati and Mohebi, 2008 [9] explored the application of ANN to study the design of short-term temperature forecasting (STTF) systems for Kermanshah city, west of Iran. They used Multi-Layer Perceptron (MLP) to model STTF systems. Their study based on training and testing MLP using ten years (1996-2006) meteorological data. Their results show that MLP network has the minimum forecasting error and can be considered as a good method to model the STTF systems.

Bilgili, 2010 [10] developed ANN model to estimate monthly mean soil temperature for the present month by using various previous monthly mean meteorological variables. The measured soil temperature and other data between years [2000..2007] at Adana meteorological station were used. The soil temperatures were measured at different depths below the ground level by the Turkish State Meteorological Service (TSMS). A feed-forward ANN was constructed with 3-layers and a BP was used for the training this ANN. The models based on the combination of input variables were constructed and the best fit input structure was investigated. The performances of ANN models in training and testing procedures were compared with the measured soil temperature values to identify the best fit forecasting model. The results show that the ANN approach is a reliable model for prediction of monthly mean soil temperature.

According to above, we need to identify the best ANN model for soil temperature estimation at any depth. The objective of this paper is to develop ANN model that can be used for best prediction of soil temperature by using various meteorological variables of any day of the year in the city of Mosul- Iraq. At the same time, we need to identify the best optimization training algorithm to train the best ANN model. The remaining of this research is as follows: section 2 includes details about different BPNN architectures. Section 3 explains materials, methodology and proposed model. Section 4 includes the results and section 5 concludes this research.

2 BACKPROPAGATION ANN

A neural network can be classified into: static and dynamic categories. Static feed forward network has no feedback elements and contain no delays like Fig.1. The output is calculated directly from the input through feed forward connections like BP, Cascade BPNN. In static network, the artificial neuron model is comprised of a summing and an activation function. The worth of each input value is assessed through synaptic weights, and then, all weighted inputs are added. To correct for a linearity assumption a distributive value of bias is added to the summing function (Eq.1). The result forms the argument of an activation function φ that acts as a filter and which yields the neuron's response as a single number (Eq.2) [11]. For hydrologic processes the procedure for a single neuron k, where input parameters j are given as time series, $x_j(t)$, can be described at each time interval as follows:

$$U_K(t) = \sum_{j=1}^{n} w_{jk}(t).x_j(t) + b_{0k}(t) \qquad (1)$$

$$Y_k\ (t) = \varphi\ (U\ _k\ (t)) \qquad (2)$$

Here $x_j(t)$ is the input value of parameter j at time-step t; $w_{jk}(t)$ is the weight assigned by neuron k to the input value of parameter j at time t; φ is a nonlinear activation function; b_k (t) is the bias of the k-neuron at time t, and $y_k(t)$ is the output signal from neuron k at time t. The process can be repeated for all entries of the time series and yields an output vector $y_{k.}$

Neurons can form layers that are fully interconnected creating networks. A typical network consists of three layers: input layer, hidden layer and output layer. The input layer refers to the available data that enter the system. The number of input layer neurons is equal to the number of parameters that contribute to the simulation. The number of hidden layers can be more than one according to the problem's complexity. Finally, the output layer returns the output vectors, which are the final responses of ANN [11]. A typical BPNN is shown in Fig.1.

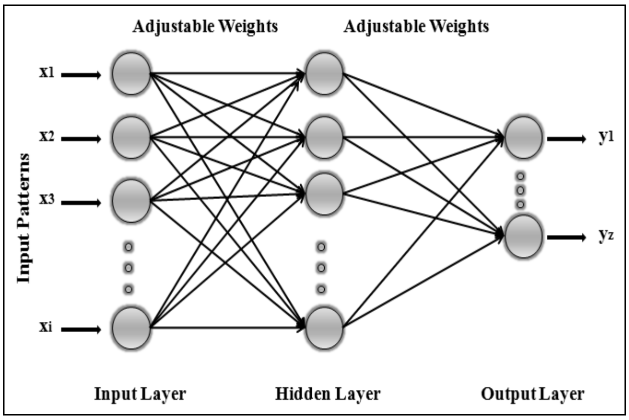

Fig.1: Typical BPNN

The learning process refers to the adjustment of weights, through which the inputs are linearly related, to minimize the error between the network's prediction and the actual response. It is an iterative procedure that adjusts the synaptic weights values when the network gaining extra knowledge after each iteration. The output y_k is compared with the target output T_k using an error function (Eq.3):

$$\delta_k = (T_k - y_k)\, y_k\, (1 - y_k) \qquad (3)$$

For the neuron in the hidden layer, the error term is given by Eq.4 [12]:

$$\delta_k = y_k\ (1 - y_k)\, \Sigma \delta_k w_k \qquad (4)$$

Where δ_k is the error term of the output layer and w_k is the weight between hidden and output layer. The error is then propagated backward from the output layer to the input layer to update the weight of each connection [12]:

$$w_{jk}(t+1)=w_{jk}(t)+ \eta\delta_k\, y_k + \alpha(w_{jk}(t)–w_{jk}(t–1)) \qquad (5)$$

Here, η is the learning rate, and the term α is called the momentum factor, which determines the effect of past weight changes on the current direction of movement. Both of these constant terms are specified at the start of the training cycle and determine the speed and stability of the network [13]--[17].

2.1 Cascade-Forward ANN

Feed-forward networks with more layers and connection might learn complex relationships more quickly like cascade-forward networks. These are similar to BPNN but include a weight connection from the input to each layer and from each layer to the successive layers. Fig.2 shows a three-layer network has connections from layer 1 to layer 2, layer 2 to layer 3, and layer 1 to layer 3. The three-layer network also has connections from the input to all three layers. The additional connections might improve the speed at which the network learns the desired relationship [13]--[17].

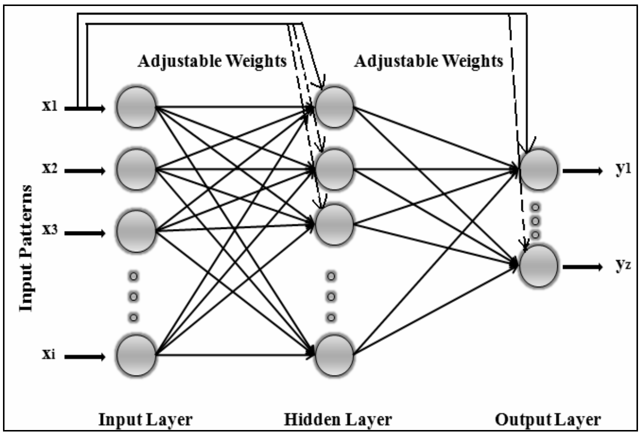

Fig. 2: Cascade-forward ANN

2.2 Recurrent Dynamic ANN

Other ANNs can learn dynamic or time-series relationships. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network. The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network with feedback connections enclosing several layers of the network. The NARX model is based on linear ARX model which is commonly used in time-series modeling, where the next value of the dependent output signal *y*(*t*) is regressed on previous values of the output signal and previous values of an independent input signal. You can implement the NARX model by using a feedforward ANN to approximate the function *f*. Fig. 3 shows where a two-layer feedforward network is used for the NARX network. The defining equation for the NARX model is:

$$Y(t) = F(y(t-1), y(t-2),\ldots..,y(t-n_y), \\ u(t-1 , u(t -2),\ldots.., u(t-n_u)) \qquad (6)$$

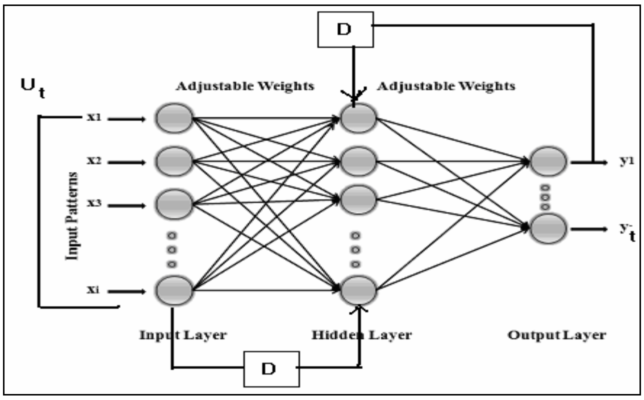

Fig.3: Recurrent Dynamic ANN

Although dynamic networks (time serious) can be trained using the same gradient-based algorithms (BP) that are used for static networks, the performance of the algorithms on dynamic networks can be quite different, and the gradient must be computed in a more complex way. The weights have two different effects on the network output. The first is the direct effect, because change in weight causes an immediate change in the output at the current time step (This first effect can be computed using standard BP). The second is an indirect effect, because some of the inputs to the layer such as $a_{(t-1)}$ are also functions of the weights. To account for this indirect effect, you must use dynamic BP to compute the gradients, which is more computationally intensive. Expect dynamic BP to take more time to train. In addition, the error surfaces for dynamic networks can be more complex than those for static networks. Training is more likely to be trapped in local minima. This suggests that you might need to train the network several times to achieve an optimal result [13]--[17].

3 MATERIALS and METHODOLOGY

According to the background in section 2, we designed and trained three layered ANN model with sigmoid transfer function for hidden layer, and linear transfer function for output layer to represent any functional relationship between inputs and outputs, if the sigmoid layer has enough neurons.

The weather data of 4 years which will be used as input to the designed ANN were taken and collected from the office of Soil and Water in Nineveh-Iraq. Table 1 shows the part of weather data (i.e. variables used to determine structure of ANN) which has measured at hours 9, 12 and 15 respectively. Another variable which we got soil temperature at soil depth equals 5, 10, 20, 30, 50 and 100cm respectively. We used also average temperature of air, sunshine and radiation. The general structure of input/outputs of the model is shown in Fig. 4.

Table 1: Metrological variables

No.	Daily meteorological variables	min	max
1	Time	9	15
2	Day	1	365
3	Avg_temperature	1	35
4	Sunshine	0	20
5	Radiation	84	770
6	Depth temperature 5	2	40
7	Depth temperature 10	3	40
8	Depth temperature 20	4	36
9	Depth temperature 30	5	36
10	Depth temperature 50	1	33
11	Depth temperature 100	1	30

3.1 The Used ANN Structure

We used the multilayer ANN to be trained by BP, which principles are based on error correction learning. When a pattern (input vector) is presented to this BPNN for the first time, it produces a random output (actual). The difference between this output and the desired compose the error (that is calculated by BP). The BP makes that the weights from output layer been the first to be adjusted and after the weights from residual layers, correcting them from back to front, with the objective of reduce the error. This process is repeated during the learning until the error become acceptable.

The structure of the proposed BPNN for temperature forecasting is shown in Fig. 4. The chosen weather data were divided into two randomly selected groups: the training group (corresponding to 70% of the patterns), and the test group (corresponding to 30% of patterns); so that the generalization capacity of network could be checked after training phase. We used the Mean Square Error (MSE) (Eq.7), determination coefficient R_square and root mean square errors (RMSE) (Eq.8) as a measure of error made by the suggested BPNN.

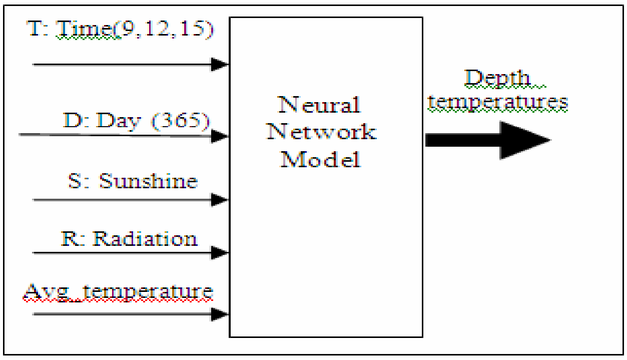

Fig. 4: Structure of Proposed Model

$$MSE = \sum_{i=1}^{n} \frac{\left(error_i\right)^2}{n} = \sum_{i=1}^{n} \frac{\left(Ti-Oi\right)^2}{n} \quad (7)$$

$$RMSE = \sqrt{\sum \frac{\left(T - O\right)^2}{n}} \quad (8)$$

Where **n** is the total number of days.

3.2. Optimization Training Algorithms for ANN models

The optimization training algorithms adjusted the ANN weights and biases with the goal to minimize the performance function. In static feed forward networks or recurrent dynamic networks, the performance function to be minimized is taken to be the mean square error, between model-predicted output and actual response. In each synapse between connected neurons the training function calculates the output error and determines the adjustments to the network's weights and bias.

In this research, five different optimization ANN training algorithms were used to train the identified best ANN model to get best results for soil temperature forecasting. These optimization training algorithms are as follows [18]:

1. Gradient descent (GD)
2. Gradient descent with momentum (GDM)

3. Gradient descent momentum and an adaptive learning rate (GDX)
4. Levenberg-Marquardt (LM)
5. Bayesian regularization (BR)

The above training algorithms can be used to train the any ANN to adjust the weights of the network to reduce errors as possible as.

4 Experimental Results

For the development of forecasting models, the total 8760 data records (4 years × 12 months × 6 depths) for each variable were collected for the period 1980-1984 for the city of Nineveh-Iraq. The data set was divided into two subsets: training and a testing data set. The training data set included a total of 6570 data records from 1980-1983, which was 70% of the total data records. For more reliable evaluations and comparisons, the models were tested with the testing data set (which was not used during the training process). The testing data set consisted of a total 2190 data records, which was 30% of the total data.

We designed various structures of BPNN forecasting models with MATLAB software to determine the optimal ANN architecture. We changed the number of neurons in the input layer.

We used different numbers of hidden layer neurons (between 10 and 35) to predict soil temperature. We used different training algorithms (BP, Cascaded BP and NARX). The different structures of forecasting models and their outputs are given in Table II. The predicted results for each model were compared statistically using three parameters: Mean Square Error (MSE), the determination coefficient (R-square) and root mean square errors (RMSE). This parameter used to see the convergence between the target (T) values and the output (O) values.

We trained and tested the models given in Table 2 to compare and evaluate the performances of ANN models. From Table 2, we can note that the best BPNN model is NARX due to its number of iterations to learn the estimation of soil temperature. The actual data and testing result of the best BPNN model (NARX) for prediction of soil temperature with different depth (Dep=5cm, Dep=10cm, Dep=20cm, Dep=30cm, Dep=50cm, Dep=100cm) are presented in Fig.5, Fig6, Fig.7, Fig.8, Fig9 and Fig.10 respectively.

Through testing we noticed that there are significant between the accuracy of the data and the data actually tested with a few errors in the prediction.

Table 2: Different structures of forecasting models

Model	Input Structure	Outputs	No. of hidden layer neurons	Epoch (No. of iterati-ons)
M1(BP)	T, D, Av	$D_{(5,10,20,30,50,100)}$	25	35
M2(BP)	T, D, Av, S	$D_{(5,10,20,30,50,100)}$	25	39
M3(BP)	T, D, Av, S, R	$D_{(5,10,20,30,50,100)}$	35	35
M4(BP)	T, D, Av, S, R	$D_{(5,10,20,30,50,100)}$	25	29
M5 (CascadeBP)	T, D, Av, S, R	$D_{(5,10,20,30,50,100)}$	15	14
M6(NARX)	T, D, Av, S, R	$D_{(5,10,20,30,50,100)}$	15	12

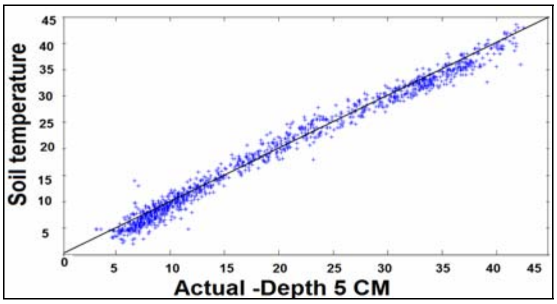

Fig.5: BPNN Prediction (Dep=5cm)

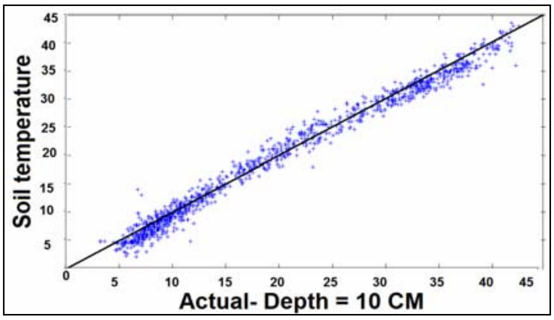

Fig.6: BPNN Prediction (Dep=10cm)

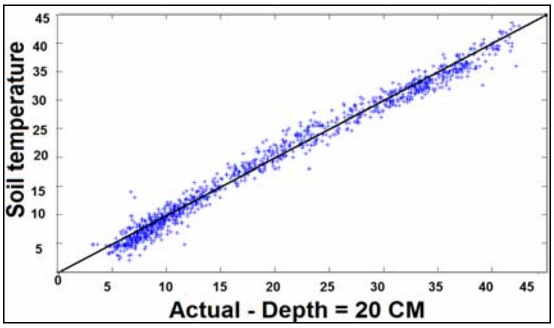

Fig.7: BPNN Prediction (Dep=20cm)

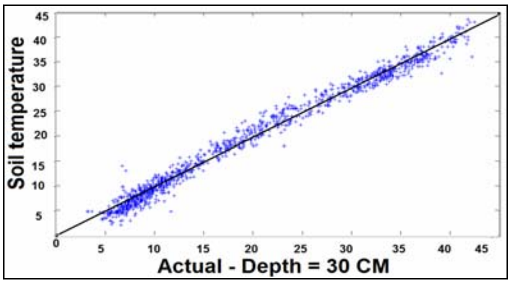

Fig.8: BPNN Prediction (Dep=30cm)

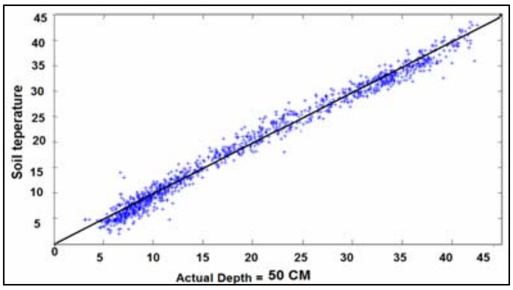

Fig.9: BPNN Prediction (Dep=50cm)

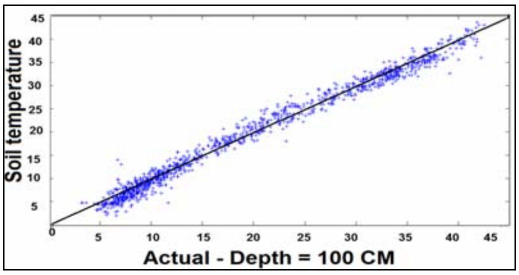

Fig.10: BPNN Prediction (Dep=100cm)

We used different architecture models of BPNN as shown previously in Table 2: M4: BP, M5: Cascade BP, and M6: NARX. During experiments, we tried to change the architecture of BPNN by changing the number of hidden layers to finally, choose the suitable BPNN model which obtain the best results of soil temperature prediction.

We note from the BPNN training process that, taking one value (factor) of weather is not enough to predict the correct soil temperature of the day. It is more effective to take many values (factors) of weather of any day of the year such as (time, average air temperature, radiation and sunshine) to predict the correct soil temperature of any day in the year.

At the same time, the number of neurons in the hidden layer has important effect on BPNN results. We trained the BPNN with different number of hidden layer neurons for the same input data. From training process, we note that the number of hidden layer neurons ranged between 15 and 25 has best effect on BPNN results.

Table 3 lists the Mean Square Error (MSE), Root Mean Square Error (RMSE) and R^2 of the prediction results of the three different trained models (M4: BP, M5: Cascade BP, and M6: NARX) with different soil depths.

Table 3: MSE, RMSE and R2 of models' prediction results

	Model	D_5	D_{10}	D_{20}	D_{30}	D_{50}	D_{100}
MSE	M4	8.574	6.496	5.631	4.861	2.811	1.695
	M5	6.096	4.961	4.17	3.485	2.08	1.193
	M6	2.051	1.254	0.986	0.515	0.752	0.412
RMSE	M4	2.928	2.549	2.373	2.205	1.677	1.302
	M5	2.469	2.228	2.042	1.867	1.441	1.092
	M6	1.432	1.119	0.993	0.718	0.867	0.641
R^2	M4	0.93	0.94	0.94	0.94	0.95	0.95
	M5	0.95	0.95	0.95	0.96	0.96	0.96
	M6	0.984	0.99	0.99	0.99	0.99	0.99

As shown in Table 3, the error rates (MSE) was gradually decreased from the surface soil down to the depths for all models (M4, M5 and M6). All the results showed that the depth of 100 gives a daily forecasting very close to the truth of the readings.

On the other hand, the value of R-square for all models (M4, M5 and M6) ranged from 0.93 to 0.98 with depth of 5cm. While the value of R-square ranged from 0.95 to 0.99 with depth of 100cm. All of the values of R-square were taken when apply time factor to the BPNN with the values: 15, 12 and 9 respectively.

After training, the testing results showed that there is a correct prediction of daily soil temperature with different depths. From testing, the model M4: BP gave a soil temperature prediction correctly by 75%. Whereas the model M5: Cascade BP gave predict correctly by 80%. While the model M6: NARX gave soil temperature prediction correctly by a large quite reached 95%. Fig.11 shows the MSE of the three models (M4, M5 and M6) and the best MSE values are for model M6 and ranged from 2.051 at depth 5cm to 0.412 at depth 100cm. At the same time, Fig. 12 shows the values of R-square for the models (M4, M5 and M6) ranged between 0.98 at depth 5cm to 0.99 at depth 100cm.

After training ANN: NARX, Table 4 shows the final values of the weights between the hidden layer and the output layer. These values can be used later to get the values of the output layer neurons for any real daily readings (for input) at any hour in any day of the year.

Table 4 can be used with the equations (9), (10) and (11) to get the correct estimation of soil temperature with different depths for any real daily reading for any day of the year and at any time.

To estimate the soil temperature of any depth for any given input reading, use the Eq.9.

$$E_{ij}=T_i\times W_{ij}+D_i\times W_{ij}+S_i\times W_{ij}+R_i\times W_{ij}+A_i\times W_{ij} \qquad (9)$$

Where $i=(1,2,3,4,5)$ for input layer neurons and $j=(1,2,3,4,\ldots\ldots..15)$ for hidden layer neurons.

Apply the Eq.10 to compute the soil temperature of all depths

$$F_{jk} = 1\ /\ (\ 1+e^{-E_{ij}}\ ) \qquad (10)$$

Finally, use Eq.11 to find final results (i.e. the correct soil temperature estimation with different depths). This is done by:

(1) Input the real measured values to Eq.9.
(2) Use the real weights which make the ANN: NARX more stable.
(3) Compute the output from Eq.10.
(4) Compute the final values of outputs for all depths and depend on weights in Table 4 using Eq.11

$$D_{T(5,10,20,30,50,100)} = \sum(F_{jk} \times W_{jk}) \qquad (11)$$

where k=1,2,3,4,5,6 for depth

As example, use the first column (W_{j1}) in Table 4 to estimate the soil temperature at depth equal 5, use the following equation:

$$D(5) = -0.20079\times Fj1 - 0.01194\times Fj1 +\ldots+ 2.370612\times Fj1$$

Table 4: Final values of weights between hidden and output layer

W_{j1}	W_{j2}	W_{j3}	W_{j4}	W_{j5}	W_{j6}
-0.2007	-0.2483	-0.3157	-0.3944	-0.3408	-0.035
-0.0119	-0.1973	-0.1540	-0.0878	0.03073	-0.038
0.1029	-0.0355	-0.0164	0.02747	0.00814	0.0310
-0.1073	-0.2951	-0.2961	-0.3051	-0.3487	-0.076
-0.1506	0.21711	0.0896	0.00872	-0.0129	-0.090
0.0601	-0.2171	-0.1960	-0.1807	-0.1435	0.0444
0.1188	0.15229	0.12583	0.10218	0.12172	0.0600
0.0023	-0.0882	-0.0674	-0.0529	-0.0453	0.0150
0.0114	0.02361	0.05755	0.02493	-0.0243	0.0306
0.0263	0.05621	0.02368	0.09365	-0.0808	-0.170
0.0401	-0.0282	-0.10373	-0.09571	-0.1196	-0.038
-0.2741	-0.21598	-0.14176	-0.10553	-0.1525	-0.085
0.0280	0.02479	0.01956	0.02038	0.01676	0.0221
0.06802	-0.14198	-0.10936	-0.09405	0.00942	0.0280
2.37061	1.019359	2.496822	3.620902	1.61798	-0.579

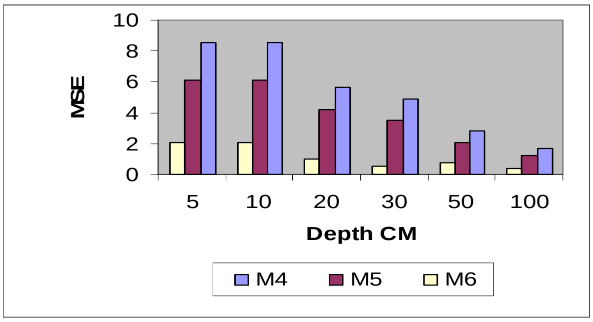

Fig.11: MSE of prediction results

According to the testing results, we noticed that the model M6: NARX is the best model that could be adopted in the prediction of deep soil temperatures with consideration of the average temperature air, radiation and sunshine at times (15, 12 and 9).

Finally, the high performance of BPNN model comes from the nature of input data, time factor and the BPNN structure (M6: NARX) including training BP which all will build a strong prediction model for soil temperature with best results.

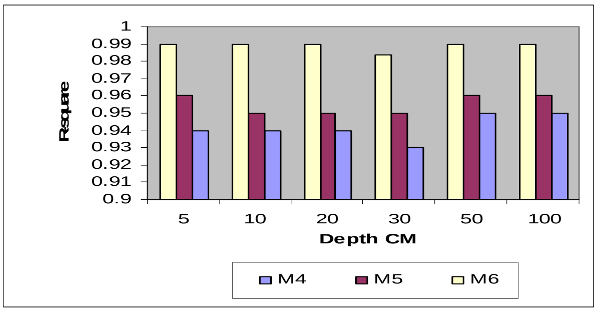

Fig.12: Values of (R^2) of prediction results

4.1. Optimal Algorithm for NARX

In this research, five different optimization ANN training algorithms (T1:GD, T2:GDM, T3:GDX, T4:LM and T5:BR) were used to train NARX ANN model (with 15 neurons in a single hidden layer) to get best results for soil temperature forecasting as shown in Table 5.

Table 5: Optimization training algorithms

	Training algorithms	Number of Neurons in hidden layer	Epoch (No. of iteration)
1	TrainGD(T1)	15	19
2	TrainGDM(T2)	15	20
3	TrainGDX(T3)	15	23
4	TrainLM(T4)	15	12
5	TrainBR(T5)	15	17

From the Table 5, we can see that TrainLM algorithm is the best and fastest training algorithm for NARX ANN. It requires only 12 iterations to convergence the NARX ANN. Table 6 shows MSE, RMSE, and R^2 for all training algorithms (T1 to T5) used for training NARX ANN.

Table 6: MSE and RMSE of NARX training algorithms

	Model	D_5	D_{10}	D_{20}	D_{30}	D_{50}	D_{100}
MSE	T1	2.066	1.289	1.015	0.770	0.674	0.439
	T2	2.925	1.301	1.054	0.702	0.745	0.510
	T3	2.075	1.293	1.023	0.721	0.692	0.454
	T4	2.051	1.254	0.986	0.515	0.752	0.412
	T5	2.056	1.276	0.995	0.764	0.623	0.422
RMSE	T1	1.437	1.135	1.008	0.82	0.88	0.662
	T2	1.71	1.14	1.03	0.86	0.84	0.714
	T3	1.44	1.14	1.012	0.85	0.83	0.674
	T4	1.432	1.119	0.993	0.718	0.867	0.641
	T5	1.434	1.13	0.998	0.874	0.79	0.65
R^2	T1	0.98	0.987	0.99	0.99	0.99	0.99
	T2	0.973	0.982	0.986	0.99	0.99	0.99
	T3	0.976	0.984	0.99	0.99	0.99	0.99
	T4	0.984	0.99	0.99	0.99	0.99	0.99
	T5	0.982	0.996	0.99	0.99	0.99	0.99

From Table 6, we can see that the smallest values of RMSE are when using the TrainLM algorithm. And the values of R^2 are between 0.99 and 0.98 for all training algorithms (T1 to T5).

We mentioned earlier in the literature studies, one study which was adopted by Bilgili [10] and it is similar to our study but with many differences. Table 7 lists the main features of our study in comparison to study adopted by Bilgili [10].

Table 7: comparisons with reference [10]

Our research	**Reference [10]**
Develop best ANN model to predict soil temperature for any day of year using various previous day variables	Develop ANN model to predict monthly soil temperature for present month using various previous monthly variables
Used BP, Cascade-forward and NARX Time Series (Nonlinear Autoregressive)	Used only BP algorithm
ANN model consisting of 5 input variables (previous day atmospheric temperature, sunshine, radiation, time, day of the year)	ANN Model consisting of 4 input variables (previous monthly soil temperature (S_{t-1}), previous monthly atmospheric temperature (T_{t-1}), depth (D), and month of the year (Mt))
Six outputs from ANN to represent soil temperature with many depths (5, 10, 20, 30, 50,100 cm) of day	Only one output from ANN to represent soil temperature of a specified depth of month.
Forecasting soil temperature for any depth (5, 10, 20, 30, 50,100 cm) of Nineveh city	Forecasting the monthly mean soil temperature of Adana city in Turkey.
R^2 is 0.99 in depth 10, 20, 30, 50 and 100cm	R^2 is 0.99 in depth 100cm
Many Optimization training algorithms were used for NARX ANN	Only one algorithm (BP)

From the Table 7, we can note that, the research [10] had good estimation of soil temperature but for one depth with specified month.

In our study, the time and day have important effect on estimation of soil temperature for any day in the year because the data were listed sequentially according the days of the year (increase and decrease in temperature degrees)

The sequentially nature of ANN: NARX with sequential data have a large effect in correct estimation of daily soil temperature with different depths (5, 10, 20, 30, 50 and 100). Therefore our ANN has 6 outputs with consideration of output weather circumstances.

At the same time, we conducted another comparison between our suggested ANN: NARX model and the model explained in reference [2]. Table 8 shows this comparison.

Table 8: Comparisons with reference [2]

	Our Research	**Reference [2]**
RMSE	Between 0.64 and 1.4	Between 0.59 to 1.82°C
R^2	Between 0.98 and 0.99	Between from 0.937 to 0.987
Depth	5 cm to 100 cm	100mm, 500mm and 1500mm
Used ANN	M6: NARX	One hidden layer BPNN
Used Factors	5 inputs: temperature, sunshine, radiation, time, day of the year.	3 inputs: daily rainfall, evaporation, day of the year
Optimization training algorithms	Train GD TrainGDM TrainGDX TrainLM TrainBR	Only one algorithm (BP)

Therefore, we can use our obtained ANN: NARX results to estimate the soil temperature for any day in the year and at any depth by considering routinely measured meteorological parameters.

5 CONCLUSION

In this study, BPNN models (M4: BP, M5: Cascade BP, and M6: NARX) were used to predict the day soil temperature for the present day by using five various previous day meteorological variables in Nineveh-Iraq. The five variables (previous day atmospheric temperature, sunshine, time and day of the year) represent the input to the ANN model.

After ANN training and testing, we noticed that, the model M4: BP gave a soil temperature prediction correctly by 75%. Whereas the model M5: Cascade BP gave predict correctly by 80%. While the model M6: NARX gave soil temperature prediction correctly by a large quite reached 95%.

From a series of BPNN exercises and according to the values of MSE, we found that, the M6: NARX model was the best model for forecasting the soil temperature for any depth (5, 10, 20, 30, 50, 100cm) of the city of Nineveh, Iraq.

The results obtained with NARX model were compared with the measured data. Errors obtained within the acceptable limits. The best result was found in the depth of 100cm. The advantage of NARX model is that, having the required various previous day meteorological variables, the day soil temperature for the present day can be predicted quickly and satisfactorily without the use of any other parameters related to soil.

Finally, to get best results for NARX ANN model for soil temperature forecasting, we used five different optimization ANN training algorithms (TrainGD, TrainGDM, TrainGDX, TrainLM and TrainBR) to train the NARX. From our results, we noticed that, TrainLM optimization algorithm is the best and fastest training algorithm for NARX ANN.

For future work, we suggest to make a comparison between the NARX ANN model of best results for estimating soil temperature with other statistical method for soil temperature: Auto-Regressive Integrated Moving Average (ARIMA). We will also get the weather data of the years 1990-2010 and then apply NARX ANN model. We will check the validity of the NARX results using other statistical techniques.

6 REFERENCES

[1] P. D. Wasserman, *Advanced Methods in Neural Computing*, 1st, John Wiley & Sons, Inc. New York, NY, USA, ISBN: 0442004613, 1993.

[2] C. C.Yang, S. O. Prasher, and G. R Mehuys, "An Artificial Neural Network to Estimate Soil Temperature," *Canadian Journal of Soil Science*, vol.77, pp.421–429, 1Dec.1997.

[3] H. Beigy and M. R Meybodi, "A learning Automata-Based Algorithm for Determination of the Number of Hidden Units for Three-layer Neural Networks," *International Journal of Systems Science*, vol. 40, no.1, pp.101–118, Jan. 2009, Taylor & Francis, Inc. Bristol, PA, USA, DOI: 10.1080/00207720802145924.

[4] G. Mihalakakou, "On Estimating Soil Surface Temperature Profiles," *Energy and Buildings*, vol.34, pp. 251-259, 2002.

[5] T. Yilmaz, A. Özbek, A. Yılmaz, and O. Büyükalaca, "Influence of Upper Layer Properties on the Ground Temperature Distribution," *Journal of Thermal Science and Technology*, vol.29, pp.43-50, 2009.

[6] A.J.Tenge, F.B.S. Kaihura, R. Lal, and B.R Singh, "Diurnal Soil Temperature Fluctuations for Different Erosion Classes of an Oxisol at Mlingano, Tanzania," *Soil and Tillage Research*, vol.49, no.3, pp.211-217, 2Dec1998.

[7] A. M. Garcia-Suarez and C. J. Butler, "Soil Temperatures at Armagh Observatory, Northern Ireland from 1904 to 2002," *International Journal of Climatology*, vol.26, pp.1075–1089, 2006, DOI: 10.1002/joc.1294.

[8] K I. Paul, P.J. Polglase, P.J. Smethurst, A.M. O'Connell, C.J. Carlyle and P.K. Khanna, "Soil Temperature under Forests: a Simple Model for Predicting Soil Temperature under a Range of Forest Types," *Agriculture and Forest Meteorology*, vol.121, pp.197-182, 2004.

[9] M. Hayati and Z. Mohebi, "Application of Artificial Neural Networks for Temperature Forecasting," *International Journal of Engineering and Applied Sciences*, vol.4, no.3, pp.164-168, 2008.

[10] M. Bilgili, "The Use of Artificial Neural Networks for Forecasting the Monthly Mean Soil Temperatures in Adana, Turkey." *Turk J Agric for 34*, T-BİTAK, DOI: 10.3906/tar-1001-593, 2010.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation* 2nd edition, Prentice-Hall Inc., 1999. http://www.statsoft.com/textbook/stneunet.html

[12] H. K. Elminir, Y.A. Azzam, and F. I. Younes, "Prediction of Hourly and Daily Diffuse Fraction Using Neural Network as Compared to Linear Regression Models," *Energy*, vol.32, pp.1513-1523. 2007.

[13] O. De Jesus and M. T. Hagan, "Backpropagation Algorithms for a Broad Class of Dynamic Networks," *IEEE Transactions on Neural Networks*, vol.18, no.1, pp.14 -27, Jan.2007.

[14] O. De Jesus and M.T. Hagan, "Backpropagation Through Time for a General Class of Recurrent Network," *in Proc. of the International Joint Conference on Neural Networks*, Washington, DC, vol.4, pp.2638–2643, 2001, ISBN: 0-7803-7044-9, DOI: 10.1109/IJCNN.2001.938786.

[15] O. De Jesus and M.T. Hagan, “Forward Perturbation Algorithm for a General Class of Recurrent Network," in *Proc. of the International Joint Conference on Neural Networks, Washington*, DC, vol.4, pp.2626–2631, 15-19July 2001, DOI: 10.1109/IJCNN.2001.938784.

[16] O. De Jesus, J.M. Horn, and M.T. Hagan, "Analysis of Recurrent Network Training and Suggestions for Improvements," in *Proc. of the International Joint Conference on Neural Networks*, Washington, DC, vol.4,pp.2632–2637,15-19July.2001, DOI: 10.1109/IJCNN.2001.938785.

[17] MathWorks, Neural Network Toolbox 7.0, MathWorks Announces Release 2010a of the MATLAB and Simulink Product Families, 2010, MathWorks, Inc. www.mathworks.com/trademarks.

[18]. Mathworks, Matlab, R2008a, Tutorial.