Academic Journal of Nawroz University (AJNU), Vol.12, No.4, 2023

https://doi.org/10.25007/ajnu.v12n4a1672

Artificial Neural Network Technique for Annual Rainfall Generation Applied to Three Selected Sites in Kurdistan Region, Iraq

Gaheen Sarma [1] and Evan Hajani [2]

1 Water Resources Department, College of Engineering, University of Duhok, Duhok, KRG – Iraq
2 Water Resources Department, College of Engineering, University of Duhok, Duhok, KRG – Iraq

ABSTRACT: Predicting rainfall is one of the more difficult tasks involved in weather forecasting. Due to extreme climate variations, it is now harder than ever to predict rainfall accurately. In the current study, an Artificial Neural Network (ANN) has been used to forecast the annual maximum rainfall (AMR) data from 1990 to 2021 at three chosen stations (i.e., Duhok, Erbil, and Sulaymaniya) in the Kurdistan region of Iraq. The Multilayer Perceptron (MLP) approach of ANN models was applied in generating and forecasting the AMR time series of the adopted stations. Model performance indicators such as model efficiency, correlation coefficient, root mean square error, and root mean absolute error were used to evaluate the performance of ANN for the annual rainfall prediction. The ANN models were used to forecast the AMR data for the upcoming five years (2022 to 2026). The study reveals that the MLP approach of the ANN models, which we have used is the most appropriate tool for forecasting the AMR data series in the three selected stations in the Kurdistan Region of Iraq for future time periods.

Keywords: Time series; Rainfall; ANN; Regression; Generation; AMR data; Kurdistan.

1. Introduction

Forecasting rainfall data is the act of making predictions about future events that have not yet happened. Predicting reliable and accurate rainfall data is critical to managing, improving, planning, and allocating water resources. A good prediction model can help to improve water resources management decisions. To this end, various deterministic and probabilistic forecasting models have been developed to help planners, managers, and decision-makers to make better and more informed decisions. Unfortunately, only very few studies have forecasted rainfall data in the Kurdistan region and Iraq. For example, Mohammed (2012) used the ANN models to predict rainfall for 5 climate stations in Iraq. Data from 1937 to 2010 were used to forecast the next ten years for each site. To simulate the prediction, two prediction methods were applied with the ANN: Bipolar Sigmoid and Hyperbolic Tangent. It revealed that forecasting for the next four years may be more accurate than doing for the next ten years.

Rainfall generation for future periods is one of the most challenging topics on earth (Gohil et al., 2019). A rainfall forecast plays a crucial role in agriculture, so a forecast of rainfall is necessary for the better economic growth of our country (Subramanya, 2008). In water resources engineering tasks, such as proper management of floods and mitigation of droughts, rainfall forecasting is crucial (Abhishek et al., 2012; Singh, 2016).

Early warnings of severe weather, are made possible by accurate rainfall predictions, which can help prevent deaths and damage caused by a natural disaster (Luk et al., 2001; El-Shafie et al., 2011). Despite considerable advancements in weather forecasting over the last few decades, accurate rainfall forecasting remains one of the biggest issues in operational hydrology (Hung et al., 2009). The process of predicting rainfall involves several forecasting methodologies by using historical rainfall data. Previous studies showed that the Artificial Neural Network (ANN) is one of the most suitable and reliable methods for the prediction of time series data. (El-Shafie et al., 2011; Nanda et al., 2013; Shaikh and Sawlani, 2017). For example, Hung et al. (2009) used ANN models for flood management and rainfall forecasting in Bangkok, Thailand. Four years of hourly data from 75 rain gauge sites were utilized to create the ANN models. The outcomes showed that ANN forecasts perform better results than predictions made by the persistent model. Khodashenas et al. (2010) conducted a study on rainfall prediction in Mashhad Synoptic Centre, Iran using monthly rainfall data from 1958 to 2008. The ANN models have been adopted to forecast the rainfall data. When statistical tests were used to evaluate the models' performance, it was found that the ANN model was suitable for forecasting rainfall.

El-Shafie et al. (2011) used two forecasting models for rainfall in Alexandria, Egypt based on rainfall data from 1957 to 2009 by using ANN models. To predict rainfall on an annual and monthly basis, both Multi-Layer Perceptron (MLR) and Feed Forward Neural Network (FFNN) models were applied. It has been found that the FFNN model performed better than the MLR model. In addition, Abhishek et a., (2012) used ANN models for forecasting the average rainfall over the Karnataka district of Udupi. They used rainfall data from 1960 to 2010 for 8 months (April to November). They examined three different algorithms: Cascaded Back Propagation, Layer Recurrent Network, and Back Propagation Algorithm (BPA). The results revealed that BPA outperforms the other networks, with high accuracy and the least MSE. Nanda et al. (2013) studied the ARIMA model as well as three types of ANN models (i.e., MLP, Functional-link ANN (FLANN), and Legendre Polynomial Equation for forecasting rainfall data in India. The results demonstrate that the ARIMA (1, 1, 1) model closely matched all of the ANN model's findings. However, FLANN offers better prediction outcomes than the ARIMA model.

Chatterjee et al. (2018) carried out rainfall forecasting in the southern part of West Bengal (India). To forecast rainfall, a hybrid neural network (HNN) model has been used. The data was obtained by the Dumdum meteorological station in West Bengal, India, between 1989 and 1995. The proposed model was compared to MLP-FFN and ANN models. The results revealed that the proposed model outperforms the MLP-FFN. Ghazvinian et al. (2020) used ANN models to predict monthly precipitation in Semnan city in Iran. they used data on the minimum and maximum temperatures, mean relative humidity, wind speed, sunshine hours, and monthly precipitation during a statistical period of 18 years (2000 to 2018). Precipitation was predicted using seven different scenarios that represented the model inputs. The sixth scenario, which took into account the minimum and maximum temperatures as well as relative humidity, wind speed, and air pressure, showed better results. Furthermore, Dada et al. (2021) adopted four ANN algorithms for predicting rainfall, including the Feed Forward Neural Network, Cascade Forward Neural Network, Recurrent Neural Network, and Elman Neural Network (ENN). Using data from the year 2015 to 2019 in India. The results revealed that ENN algorithms are accurate, powerful, and dependable algorithms that may be used to predict rainfall on a daily, monthly, or annual basis.

Furthermore, Al-Salihi et al. (2013) used the ANN models to predict average monthly rainfall data. Backpropagation algorithms were used to build the neural networks. The data was used from the year 1970 to 2010 from four cities in Iraq: Mosul, Baghdad, Rutba, and Basra. According to statistical results, the best model was discovered in the Rutba station during the month of December. The findings clearly demonstrate that the ANN models created are appropriate for forecasting monthly rainfall data and provide extremely high accuracy of rainfall estimation at the selected stations in Iraq. Also, Yahya and Seker (2019) developed a model using ANN, including radial basis function, fuzzy c-means, and nonlinear autoregressive network, to forecast rainfall, wind speed, humidity, temperature, and sunshine in Mosul city in Iraq using data from the years 1972 to 2017. The performance accuracy of the model provides extremely close predicted results with very small statistical errors for the forecasted years from 2018 to 2050. In another study, Murad and Salih (2020) evaluated three models, including ANN, Naïve Bayes, and Support Vector Machines (SVM) to predict rainfall in Sulaymaniyah City using data from the year 2013 to 2018. The results demonstrated that SVM with a performance of 91.57%, is the best model to predict rainfall.

The aim of the current study is to determine the ability to forecast the AMR data series based on ANN models depending on the multi-layer perceptron (MLP) neural network model in the three selected sites in the Kurdistan Region, Iraq (i.e., Duhok, Erbil, and Sulaymaniya). Several statistical tests were applied to select the best ANN model for forecasting AMR at three selected sites. Furthermore, the scope of the data covered, the most recent statistical methodology used, the station selection, and the inclusion of a number of previously unpublished AMR indices are additional ways that this study differs from previous studies.

2. Location of Study Area and Data

The data for annual maximum rainfall (AMR) of three stations (i.e., Duhok, Erbil, and Sulaymaniya) over the Kurdistan region, of Iraq, which is bounded by latitudes 35° to 37° N and longitudes of 42° to 45° E as shown in Figure 1 have been used for the present study. A 24-hr rainfall dataset for the three adopted stations was provided by the Kurdistan region’s Meteorological and Seismological department. It consists of the AMR from the year 1990 to 2021 (32 years). Table 1 shows the statistical information of the three adopted areas used in the current study.

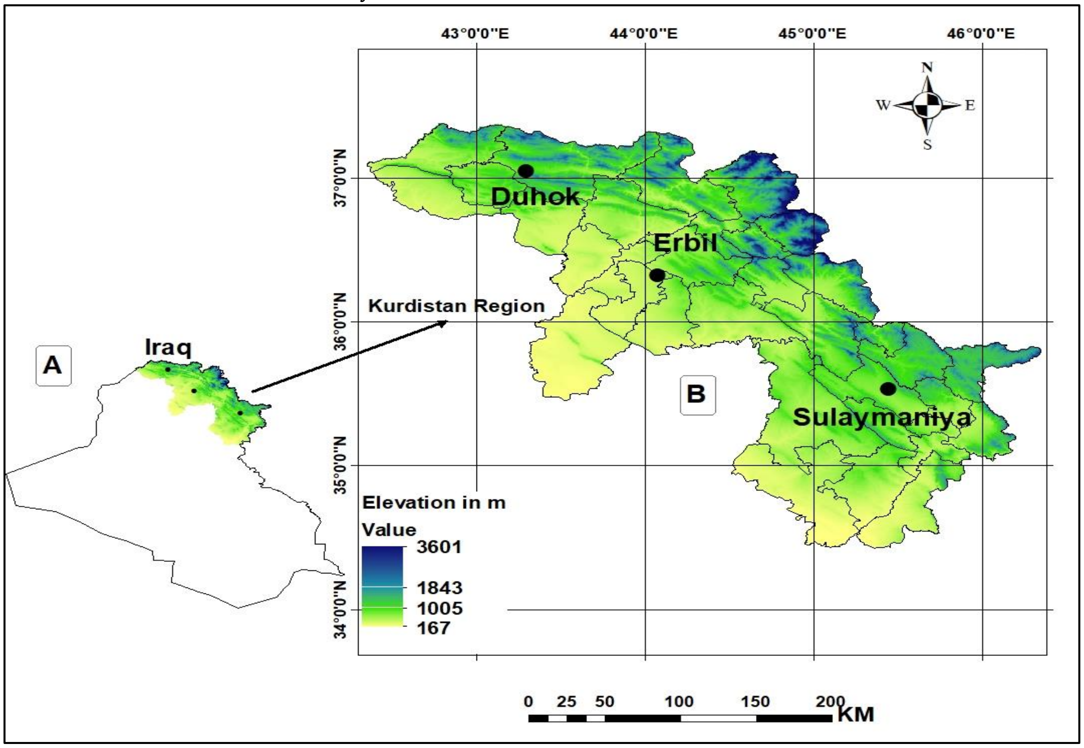

Figure 1. The location map of the study area.

Table 1: The statistical description of Duhok, Erbil, and Sulaymaniya stations.

Station	Mean	Min	Max	Std. D.	C.V.
Duhok	59.622	24.200	150.000	26.822	44.987
Erbil	48.179	23.900	103.900	17.212	35.726
Sulaymaniya	62.998	36.800	131.800	23.306	36.995

3. Methods

3.1. Artificial Neural Network (ANN) Model

An ANN consists of a set of connected artificial neurons which has the ability to store and make use of experience-based knowledge, which is a natural property. The ANNs are a class of models inspired by biological neural networks (the central nervous systems of animals, particularly the brain) that are used to approximate or estimate functions that can be affected by a big number of inputs and are typically unknown (Luk et al., 2001). Their development is driven by the following guidelines (ASCE, 2000):

i. Nodes, also known as units, cells, or neurons, are several single elements where the processing of information takes place.
ii. Through connected links, signals are sent from one node to another.
iii. Each connected link has a weight that corresponds to its connection strength.
iv. To identify its output signal, each node generally applies a nonlinear modification known as an activation function to its net input.

The procedure for applying the ANN model is demonstrated in Figure 2.

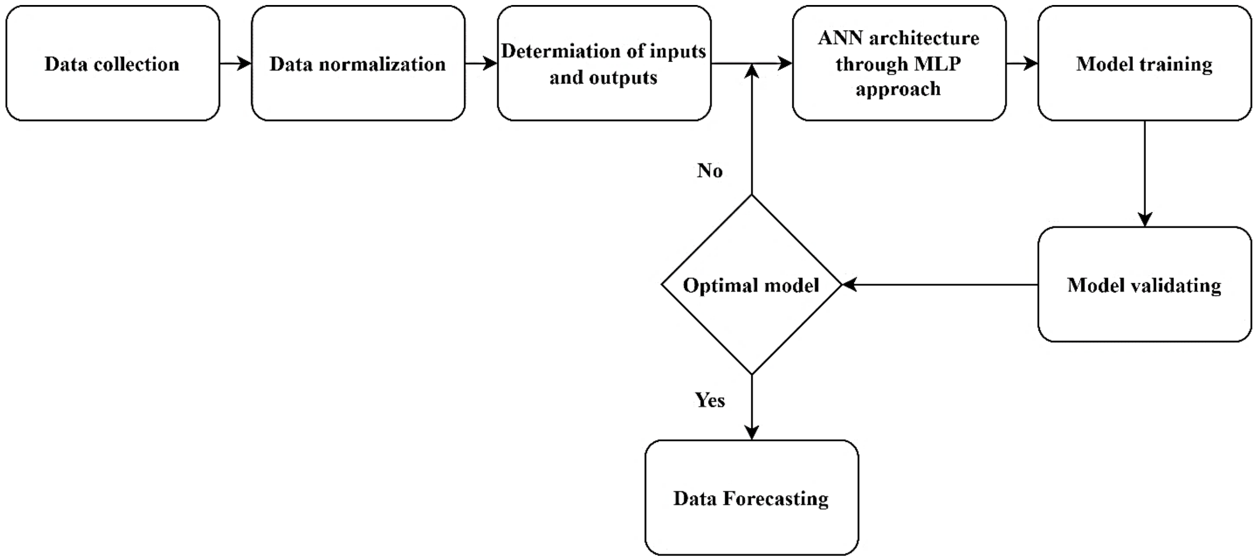

Figure 2: Flowchart of the ANN model.

3.2. Architectures of Artificial Neural Networks

A typical ANN is composed of a number of neurons that are arranged in a particular way and generally consist of three different neuron layers: input layer, hidden layer, and output layer as shown in Figure 3. Artificial neurons in one layer are completely or partly connected to artificial neurons in the next layer. It is also possible to make a feedback connection to the previous layer (Engelbrecht, 2007).

The general definition of three layers in Artificial Neural Networks:

i. Input layer: This layer receives information (data), measurements, features, or signals from the external world. The input layer does not analyze the incoming data and just forwards it to nodes in the hidden layer.
ii. Hidden, invisible, or intermediate layers: These layers consist of neurons and are placed in the middle of the input layer and output layer. It is important for performing computations and transforming the information from the input layer to the output layer.
iii. Output Layer: This layer also consists of one or multiple neurons and is in charge of completing computations and presenting the final network results.

3.3. Procedure for Modeling ANN

The following procedures must be followed when using the ANN modeling approach for forecasting (ASCE, 2000):

a) Data Pre-Processing: The pre-processing process applies to both the input and target vectors in the time series. This ensures that the output of the network always falls within the applied preprocess range. Data pre-processing in this study will only be used when considered required. At first, all inputs and outputs will be normalized by using the Box-Cox transformation method to have a mean of 0 and a standard deviation of 1.
b) Determination of ANN inputs and output: The inputs include the critical information needed to build the data generation relationship, whereas the outputs are what we need from the ANN model. The number of inputs can be large. In this case, the number of input nodes is defined directly by the number of lags (n) (Kingston, 2006).
c) Modelling of the ANN Architecture: The architecture of the ANN model for rainfall forecasting must be constructed after data pre-processing. (Hagan, 1997). The most common method for forecasting rainfall is a multi-layered feed-forward network. When creating network architecture. The architecture of a multilayer perceptron artificial neural network is demonstrated in Figure 3, which includes an input layer, a hidden layer, and an output layer of neurons. These three layers are connected by connections with weight, which denotes their strength. Thus, there are two different sets of weights: the weights of the input-hidden

layer (wi.j) and the weights of the hidden-output layer (wj,k). For example, one hidden layer perceptron network's input-output mapping demonstrated in Figure 3 may often be expressed as:

$$y_k = f_O\left[\sum_{j=1}^{m}\left(w_{j,k} * f_{Hj}\left(\sum_{i=1}^{n}\left(w_{i,j} * x_i\right) + b_j\right)\right) + b_k\right] \quad (1)$$

Where yk is the output variable, xi is the input variable, n is the nodes number in the input layer, m is the nodes number in the hidden layer, k is the nodes number in the output layer, wi,j is input-hidden weights, wj,k is hidden-output weights, bj is the bias of the hidden layer, bk is the bias of the output layer, fH is the activation function of the hidden layer, and fO is the activation function of the output layer. The following tasks have been performed: i. Choosing the essential input and output nodes, ii. Choosing the number of hidden layers and their nodes by using a trial-and-error procedure to select the optimum number of hidden layer nodes. and iii. Choosing the nodes' activation function.

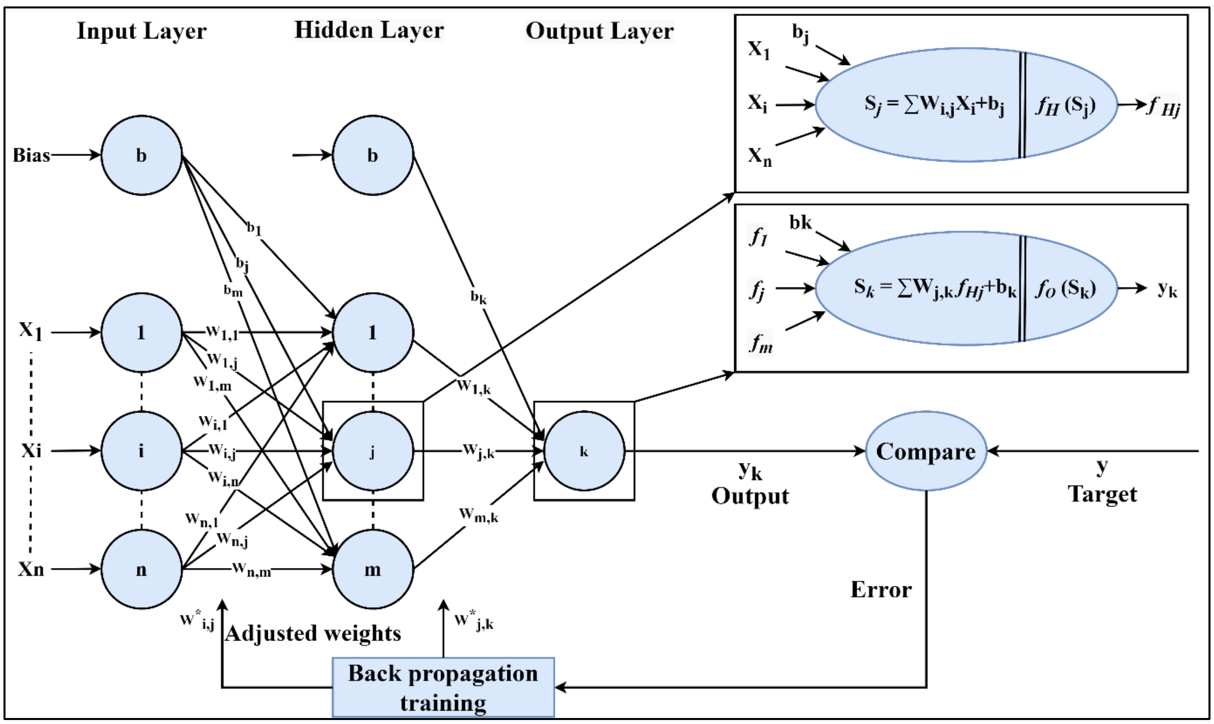

Figure 3: The MLP structure.

d) ANN Model Training: Training a neural network includes adjusting the network's weights and biases to achieve better network performance as specified by the network performance function (Singh, 2016). The performance of neural networks is highly influenced by the optimization technique chosen to calculate weight adjustments (Kassem et al. 2019).
e) ANN Model Testing: After training is finished, network performance may be examined to see whether any adjustments to the training step, network architecture, or data sets are required (Singh, 2016). However, the model must first be verified before it can be used to make predictions or simulate data. This is often done by assessing the model's performance or generalizability when applied to a separate set of validation data (a hold-out sample), using the performance criteria established. (Kingston, 2006).
f) Measurements of the reliability of the forecasting: In this study, to assess the forecasting ANN model, the relative error (RE), root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) have been adopted, as follows:

$$RE = \sum_{t=1}^{n} \frac{X_t - \hat{X}_t}{X_t} \quad (2)$$

$$RMSD = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(X_t - \hat{X}_t\right)^2} \quad (3)$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|X_t - \hat{X}_t\right| \quad (4)$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|X_t - \hat{X}_t|}{X_t} * 100 \quad (5)$$

4. The Results and Discussion

4.1. Trends and Variability in AMR data

Graphically, as shown in Figure 4, a linear regression trend line was adopted to show evidence of the AMR data variability and trend in the dataset for the three adopted rainfall stations (i.e., Duhok, Erbil, and Sulaymaniya). It can be observed in Figure 4 that in all three stations, there is a slight change in the linear trend line of the AMR, and rainfall is erratic and varies with time over the 31-year period between 1990 and 2021. The Duhok and Erbil stations have a slightly decreasing trend (negative linear trend line) of AMR data, while the Sulaymaniya station has a slightly increasing trend (positive linear trend line) of AMR data. The total annual rainfall change was computed on the fitted regression line to show the changes in AMR for each of the adopted rainfall stations. The results are -0.215 mm/year in Duhok station, -0.044 mm/year in Erbil station, and 0.448 mm/year in Sulaymaniya station.

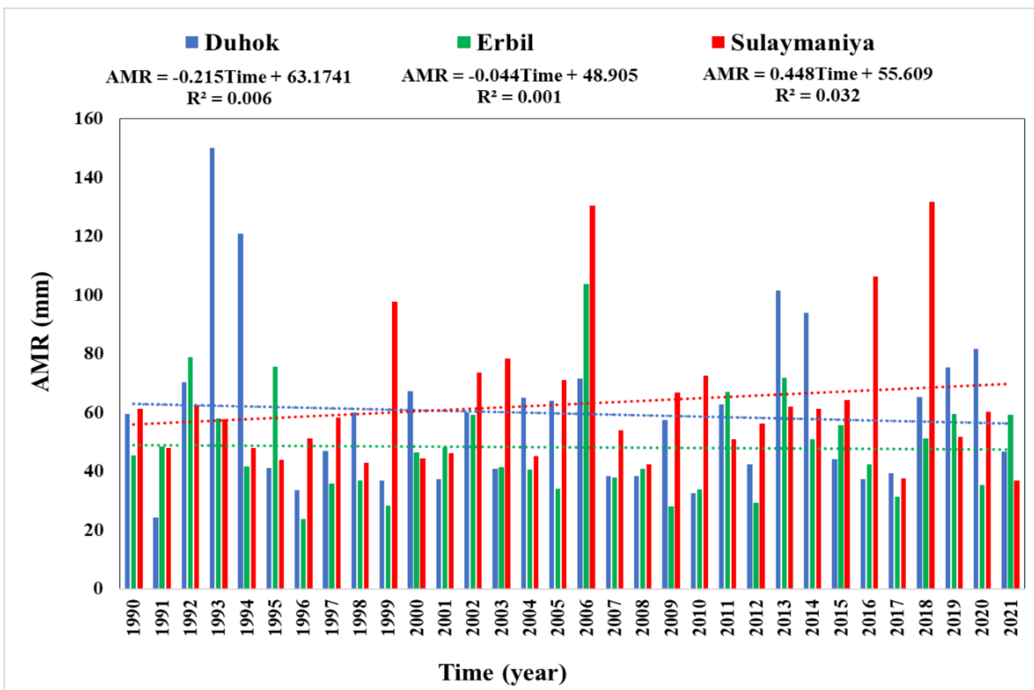

Figure 4: Long-term variability in AMR (mm) for the three adopted stations (1990-2021).

4.2. Normalize the AMR data series

Normalization is done by using the Box-Cox method (Box and Cox, 1964). The forms of the Box-Cox transformation are given by:

$$X(\lambda) = \frac{(X^{\lambda}-1)}{\lambda} \quad for\ \lambda \neq 0 \quad (6)$$

$$X(\lambda) = \log(X) \quad for\ \lambda = 0 \quad (7)$$

Where the range of lambda (λ) is -5 to 5. We take into account all possible values and choose the one that produces the best approximation of a normal distribution curve for your data (Chedded, 2020). In the current study, for the three adopted stations, the non-normal AMR data were transformed using the Box-Cox method to conform to a normal distribution, as shown in Table 2.

Table 2: The transformed function depends on the Lambda value (λ)

Station	λ: Transformed function
Duhok	0: Log (AMR)
Erbil	-0.5: 1/Sqrt (AMR)
Sulaymaniya	-1: 1/(AMR)

4.3. The application of the ANN Model for the generation of AMR data

1. Inputs and Outputs variables

As has been explained in section 3, the number of input nodes of the ANN models is directly determined by the number of lagged (n) observations of the dependent variable, which was used for forecasting the next value. It should be noted that the AMR data is used as the dependent variable in the current study. After conducting a number of lag trials, it was found that lag 6 was the optimal lag for the AMR data at the three adopted stations (i.e., Duhok, Erbil, and Sulaymaniya). In order to construct the future time series, the input layer will take 6 nodes into account for each of the three adopted stations.

2. Model Architecture

One of the most crucial and challenging steps in the creation of ANN models is determining the network architecture. As mentioned in section 3, it is a selection of the number of hidden layers and the number of nodes in each of these layers. In this study, the MLP was used, and an automatic architecture was chosen to construct a network with one hidden layer. The automatic architecture selection computed the best number of nodes in the hidden layer for the three stations that were used, as shown in Table 3. For the hidden layer, automatic architecture selection uses one of the hyperbolic tangent and sigmoid default activation functions, while one of the identities, SoftMax, hyperbolic tangent and sigmoid activation functions is selected for the output layer.

Table 3: Architectures of the best MLP model using AMR data series three adopted stations

Stations	Input layer nodes	Hidden layer nodes	Output layer nodes
Duhok	6	6	1
Erbil	6	6	1
Sulaymaniya	6	7	1

3. Generating the AMR Time Series by Using the ANN Model

The best ANN models, after many trials (more than 100), were found to be a model with hyperbolic tangent and SoftMax activation functions between the input and hidden layer, for MLP type, and identity activation function between the hidden and output layer as shown in Table 4 for AMR time series of Duhok, Erbil and Sulaymaniya stations. Figure 5 shows the MLP neural network diagrams for the AMR data model for the three stations. These diagrams show connections between input layer nods, hidden layer nods, and an output layer with their biases. In these network diagrams (Figure 5), the blue line indicates positive weights, while the red line indicates negative weights. the negative weights, and the positive weights. It should be noted that when the weight associated with a feature is positive (as in Figure 5), it implies that there is a direct relationship between that feature and the target value, and when the weight associated with the feature is negative, it implies that there is an inverse relationship between the feature and the target value.

Table 4: The MLP-Network information for the AMR data series at Duhok, Erbil, and Sulaymaniya stations

Network Information			Stations		
			Duhok	Erbil	Sulaymaniya
Input Layer	Covariates	1	Input (1)	Input (1)	Input (1)
		2	Input (2)	Input (2)	Input (2)
		3	Input (3)	Input (3)	Input (3)
		4	Input (4)	Input (4)	Input (4)
		5	Input (5)	Input (5)	Input (5)
		6	Input (6)	Input (6)	Input (6)
	Number of Units		6	6	6
	Rescaling Method for Covariates		Standardized	Standardized	Standardized
Hidden Layer(s)	Number of Hidden Layers		1	1	1
	Number of Units in Hidden Layer		6	6	7
	Activation Function		Hyperbolic tangent	Hyperbolic tangent	Hyperbolic tangent
Output Layer	Dependent Variables	1	Output	Output	Output
	Number of Units		1	1	1
	Rescaling Method for Scale Dependents		Standardized	Standardized	Standardized
	Activation Function		Identity	Identity	Identity
	Error Function		Sum of Squares	Sum of Squares	Sum of Squares

a. Excluding the bias unit

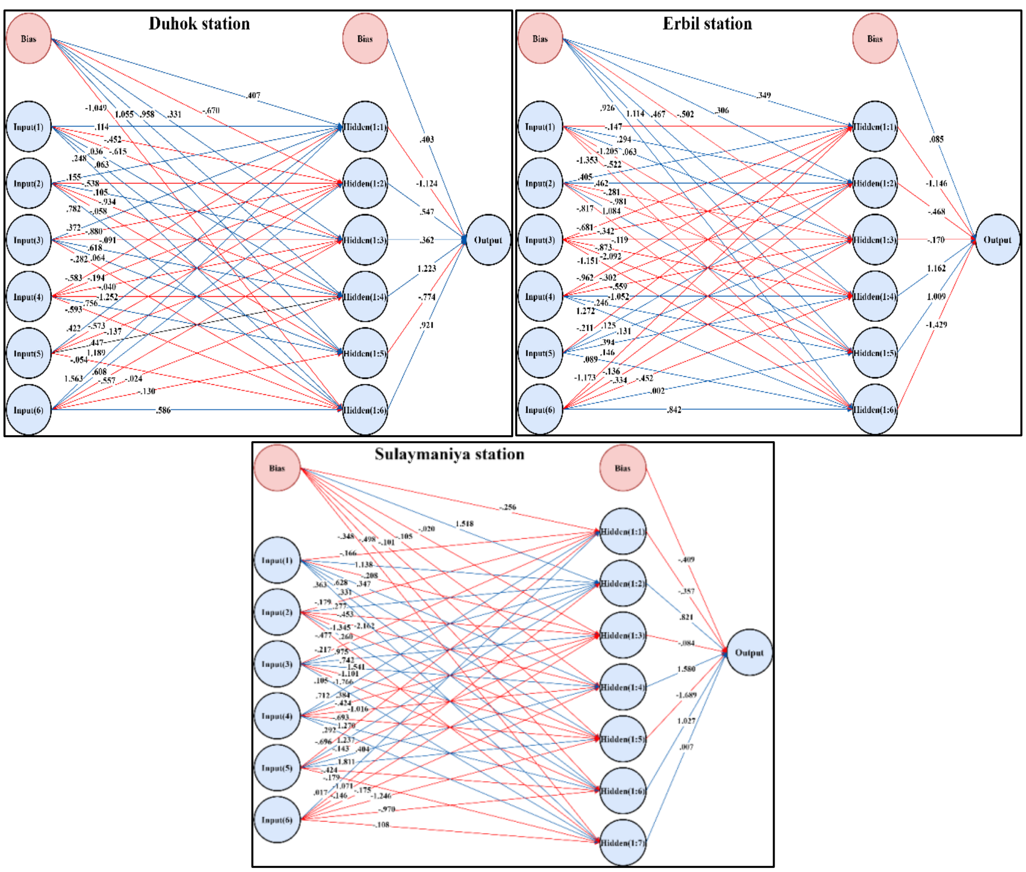

Figure 5: MLP-network diagrams for AMR data series of the Duhok, Erbil, and Sulaymaniya stations.

The weights values between input-hidden layers and hidden-output layers for MLP networks for AMR data of Duhok, Erbil, and Sulaymaniya stations were determined and are displayed in Tables 5, 6, and 7 respectively.

Table 5: MLP-network bias and weight matrices for the AMR data series at Duhok station

Parameter Estimates								
Predictor		Predicted						
		Hidden Layer						Output Layer
		Hidden (1:1)	Hidden (1:2)	Hidden (1:3)	Hidden (1:4)	Hidden (1:5)	Hidden (1:6)	Output
Input Layer	(Bias)	0.407	-0.670	0.331	0.958	1.055	-1.049	
	Input (1)	0.114	-0.452	-0.615	0.036	0.063	0.248	
	Input (2)	0.155	-0.538	0.105	-0.934	-0.058	0.782	
	Input (3)	0.372	-0.880	-0.091	0.618	0.064	-0.282	
	Input (4)	-0.583	-0.194	-0.040	-1.252	0.756	-0.593	
	Input (5)	0.422	-0.573	-0.137	0.447	1.189	-0.054	
	Input (6)	1.563	0.608	-0.557	-0.024	-0.130	0.586	
Hidden Layer	(Bias)							.403
	Hidden (1:1)							-1.124
	Hidden (1:2)							0.547
	Hidden (1:3)							0.362
	Hidden (1:4)							1.223
	Hidden (1:5)							-0.774
	Hidden (1:6)							0.921

Table 6: MLP-network bias and weight matrices for the AMR data series at Erbil station

Parameter Estimates								
Predictor		Predicted						
		Hidden Layer						Output Layer
		Hidden (1:1)	Hidden (1:2)	Hidden (1:3)	Hidden (1:4)	Hidden (1:5)	Hidden (1:6)	Output
Input Layer	(Bias)	0.349	0.306	-0.502	0.467	1.114	0.926	
	Input (1)	-0.147	0.294	0.063	-1.205	-0.522	-1.353	
	Input (2)	0.405	0.462	-0.281	-0.981	1.084	-0.817	
	Input (3)	-0.681	-0.342	-0.119	-0.873	-2.092	-1.151	
	Input (4)	-0.962	-0.302	-0.559	-1.052	0.246	1.272	
	Input (5)	-0.211	0.125	0.131	0.394	0.146	0.089	
	Input (6)	-1.173	-0.136	-0.334	-0.452	0.002	0.842	
Hidden Layer	(Bias)							.085
	Hidden (1:1)							-1.146
	Hidden (1:2)							-0.468
	Hidden (1:3)							-0.170
	Hidden (1:4)							1.162
	Hidden (1:5)							1.009
	Hidden (1:6)							-1.429

Table 7: MLP-network bias and weight matrices for the AMR data series at Sulaymaniya station

Parameter Estimates									
Predictor		Predicted							
		Hidden Layer							Output Layer
		Hidden (1:1)	Hidden (1:2)	Hidden (1:3)	Hidden (1:4)	Hidden (1:5)	Hidden (1:6)	Hidden (1:7)	Output
Input Layer	(Bias)	-0.256	1.518	-0.020	-0.105	-0.101	-0.498	-0.348	
	Input (1)	-0.166	1.138	-0.208	0.347	0.628	0.331	0.363	
	Input (2)	-0.179	0.277	-0.453	-2.162	-1.345	0.260	-0.477	
	Input (3)	-0.217	0.975	0.742	1.541	-1.101	-1.766	0.105	
	Input (4)	0.712	0.384	-0.424	-1.016	-0.693	1.270	0.292	
	Input (5)	-0.696	1.237	-0.143	0.404	1.811	-0.424	-0.179	
	Input (6)	0.017	-1.071	-0.146	-0.175	-1.246	-0.970	-0.108	
Hidden Layer	(Bias)								-0.409
	Hidden (1:1)								-0.357
	Hidden (1:2)								0.821
	Hidden (1:3)								-0.084
	Hidden (1:4)								1.580
	Hidden (1:5)								-1.689
	Hidden (1:6)								1.027
	Hidden (1:7)								0.007

In the current study, the accuracy of the models for the purposes of generation and forecasting was examined by comparing the 32-year (1990-2021) AMR data generated by ANN models with the time series data that had been recorded for the three adopted stations. The coefficient of determination (R2) and four statistical error tests, namely RE, MAE, MAPE, and RMSE (section 3), was used as a base for this comparison. Figure 6 shows the plot between the recorded and predicted values using the ANN model on the translated AMR data series for the three adopted stations, indicating that the predicted values follow the recorded data closely enough. Figure 7 shows a linear regression trend line for the adopted period (1990 to 2021) and forecasted period (2022 to 2026) to show evidence of the AMR data variability and trend in the dataset at the three adopted rainfall stations (i.e., Duhok, Erbil, and Sulaymaniya).In addition, in Table 8, it can be seen that the higher values of R^2 with lower RE, MAE, MAPE, and RMSE indicate a high degree of agreement between the data generated by the ANN model and those that were recorded, which reflects higher accuracy of the MLP neural network model for each adopted station. Finally, the MLP approach of the ANN model is applied to generate AMR data values for future periods starting from 2022 to 2026 (five years), as shown in Table 9.

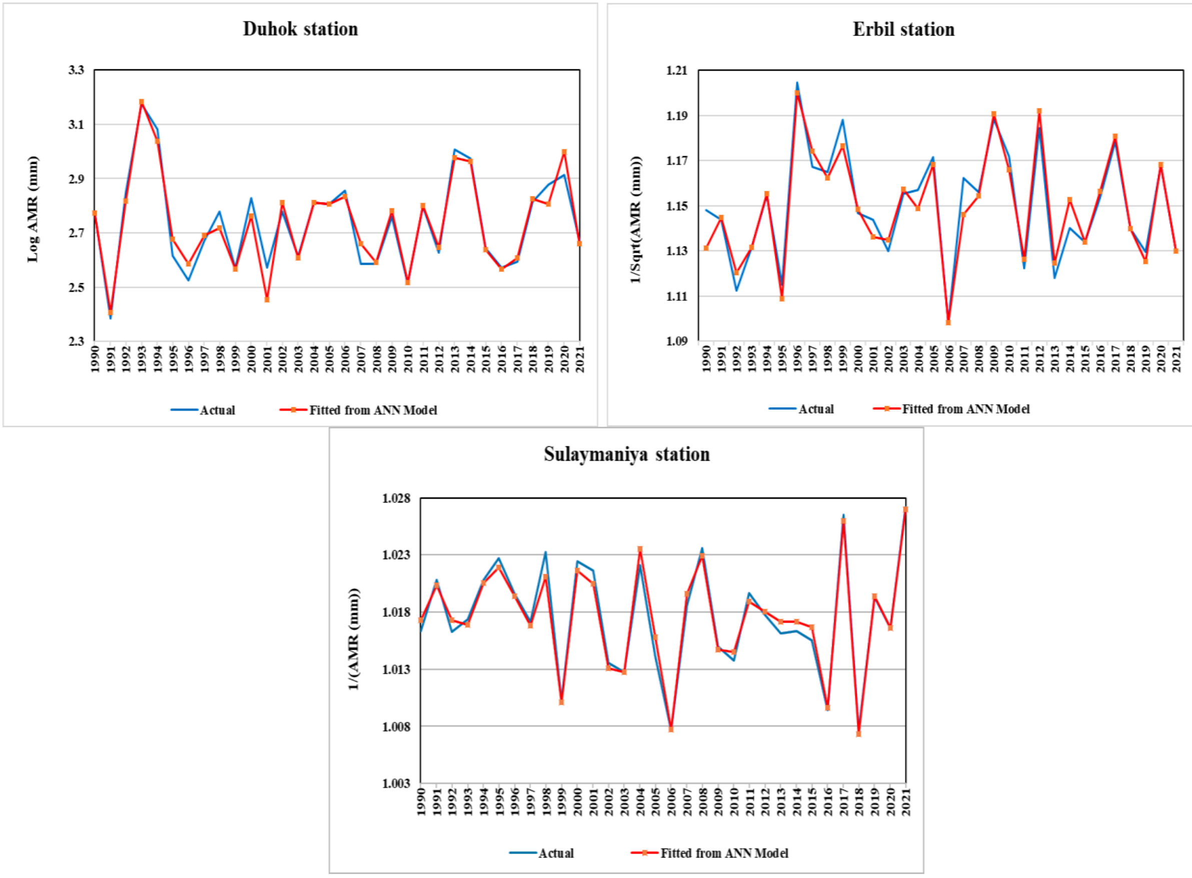

Figure 6: Time series data was generated using the ANN model for the Duhok, Erbil and Sulaymaniya stations; the actual data is given in blue color and the red line color corresponds to the fitted values.

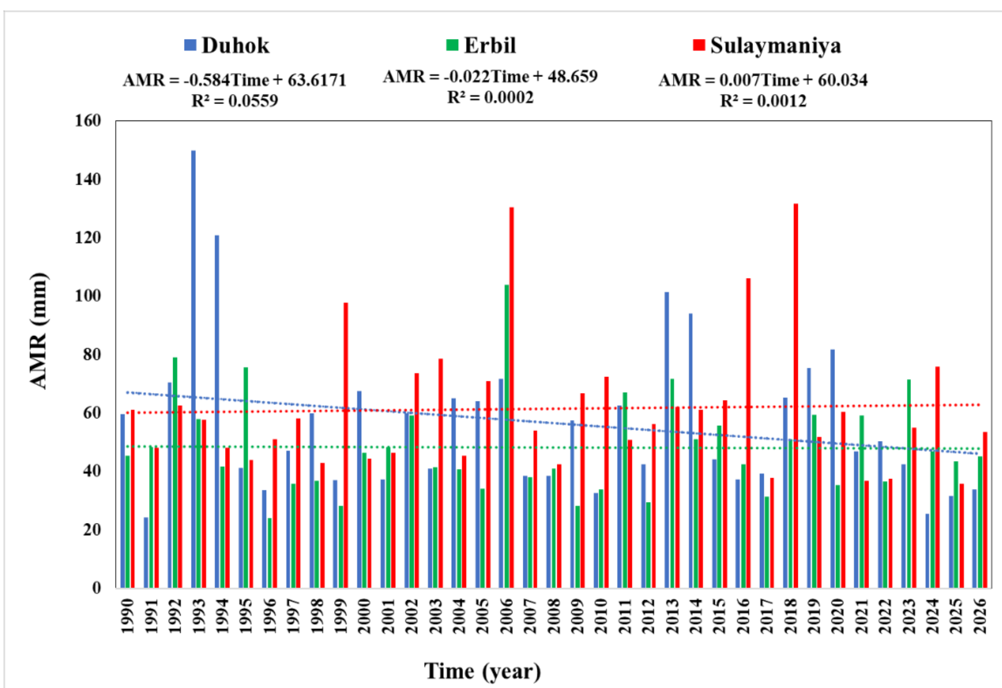

Figure 7: Long-term variability in AMR (mm) for adopted period (1990 – 2021) and forecasted period (2022 - 2026).

Table 8: Results R^2, average RE (%), MAE, MAPE (%), and RMSE between recorded AMR data and fitted AMR data by ANN model.

Stations	R^2	RE (%)	MAE	MAPE (%)	RMSE
Duhok	0.99	0.025	<0.001	0.033	0.041
Erbil	0.93	0.065	<0.001	0.012	0.006
Sulaymaniya	0.95	0.001	<0.001	0.001	<0.001

Table 9: Time series data forecasted using the ANN model from 2022 to 2026.

Stations	2022	2023	2024	2025	2026
Duhok	50.304	42.560	25.456	31.543	33.884
Erbil	36.554	71.334	46.785	43.397	45.225
Sulaymaniya	37.453	54.945	75.758	35.714	53.476

5. Conclusion

In this study, an Artificial Neural Network (ANN) model was applied to examine the pattern, distribution, and forecasting of the annual maximum rainfall (AMR) data at three selected stations (Duhok, Erbil, and Sulaymaniya) in the Kurdistan region of Iraq using 32 years of 1990 to 2021 rainfall data. The Multilayer Perceptron (MLP) approach of ANN models was applied in generating and forecasting the AMR time series of the adopted stations. The Box-Cox transformation method was used to transform non-normal AMR data to meet a normal distribution. To assess the forecasting model, the relative error (RE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) have been adopted. The results indicate that the MLP model is an appropriate model for the AMR data series for the three adopted stations. The predicted AMR data from the MLP model were compared with the observed AMR data to determine prediction precision. The MLP models were formed for each station, and it has been concluded that the best models of artificial neural networks were MLP-ANN (6,6,1) for Duhok and Erbil stations and MLP-ANN (6,7,1) for Sulaymaniya station. The results revealed that the RE, RMSE, MAE, and MAPE tests between the observed and predicted values are less than 4% (close to zero), which satisfies the criteria for forecast accuracy as mentioned in a study done by Kim and Kim 2016. Therefore, the upcoming AMR data can be forecasted using the MLP of the ANN models.

6. References

[1] Abhishek K., Kumar A., Ranjan R., and Kumar S. (2012). A Rainfall Prediction Model Using Artificial Neural Network. In 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012).

[2] Abhishek K., Singh M. P., Ghosh S., and Anand A. (2012). Weather Forecasting Model Using Artificial Neural Network. Procedia Technology, 4, 311-318.

[3] Al-Salihi A. M., Al-Lami A. M., and Mohammed A. J. (2013). Prediction Of Monthly Rainfall for Selected Meteorological Stations in Iraq Using Back Propagation Algorithms. J Environ Sci Technol, 6(1), 16-28.

[4] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000). Artificial neural networks in hydrology. II: Hydrologic applications. Journal of Hydrologic Engineering, 5(2), 124-137.

[5] Chatterje S., Datta B., Sen S., Dey N., and Debnath N. C. (2018). Rainfall Prediction Using Hybrid Neural Network Approach. In 2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom).

[6] Dada D., Yakubu J., and Oyewola D. (2021). Artificial Neural Network Models for Rainfall Prediction. European Journal of Electrical Engineering and Computer Science. 5(2).

[7] El-Shafie A. H., El-Shafie A., El-Mazoghi H. G., Shehata A., and Taha M. R. (2011). Artificial Neural Network Technique for Rainfall Forecasting Applied to Alexandria, Egypt. International Journal of Physical Sciences, 6(6), 1306-1316.

[8] Engelbrecht A. P. (2007). Computational Intelligence: An introduction. John Wiley & Sons, Ltd.

[9] Ghazvinian H., Bahrami H., Ghazvinian H., and Heddam S. (2020). Simulation Of Monthly Precipitation in Semnan City Using ANN Artificial Intelligence Model. Journal of Soft Computing in Civil Engineering, 4(4), 36-46.

[10] Gohil H., Dodiyar M., Garasiya S., and Raval K. (2019). A Review on Rainfall Forecasting. International Journal of Pure and Applied Research in Engineering and Technology, 7 (6), 137-140.

[11] Hagan M. T., Demuth H. B., and Beale M. (1997). Neural Network Design. PWS Publishing Co.

[12] Hung N. Q., Babel M. S., Weesakul S., and Tripathi N. K. (2009). An Artificial Neural Network Model for Rainfall Forecasting in Bangkok, Thailand. Hydrology and Earth System Sciences, 13(8), 1413-1425.

[13] Kassem A. A., Raheem A. M., Khidir K. M., and Alkattan M. (2020). Predicting of Daily Khazir Basin Flow Using SWAT and Hybrid SWAT-ANN Models. Ain Shams Engineering Journal, 11(2), 435-443.

[14] Khodashenas S. R., Khalili N., and Davari K. (2010). Monthly Precipitation Prediction by Artificial Neural Networks (Case study: Mashhad synoptic station). NOVATECH 2010.

[15] Kingston G. B. (2006). Bayesian Artificial Neural Networks in Water Resources Engineering, Adelaide, Australia: Computer and Mathematical Sciences. PhD. Thesis, Faculty of Engineering, School of Civil and Environmental Engineering, The University of Adelaide.
[16] Kim S. and Kim H. (2016). A New Metric of Absolute Percentage Error for Intermittent Demand Forecasts. International Journal of Forecasting, 32(3), 669-79.
[17] Kriesel D. (2005). Neural Networks Used for Prediction. D. Kriesel, A Brief Introduction to Neural Networks, 181-182.
[18] Luk K. C., Ball J. E., and Sharma A. (2001). An Application of Artificial Neural Networks for Rainfall Forecasting. Mathematical and Computer modelling, 33(6-7), 683-693.
[19] Mohammed B. A. R. (2012). Application of ANN Technique for Rainfall Forecasting Over Iraq. International Journal of Energy and Environment, 3(6), 961-966.
[20] Murad S. H., and Salih Y. M. M. (2020). Comparable Investigation for Rainfall Forecasting Using Different Data Mining Approaches in Sulaymaniyah City in Iraq. International Journal of Advances in Life Science and Technology, 4(1), 11-18.
[21] Nanda S. K., Tripathy D. P., Nayak S. K., and Mohapatra S. (2013). Prediction Of Rainfall in India Using Artificial Neural Network (ANN) Models. International Journal of Intelligent Systems and Applications, 5(12), 1.
[22] Palit A. K., and Popovic D. (2006). Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications. Springer Science and Business Media.
[23] Singh A. K. (2016). Rainfall Forecasting Using Artificial Neural Network (Doctoral dissertation).
[24] Subramanya K. (2008). Engineering Hydrology. 7 West Patal Nagar. New Delhi, 110(008).
[25] Yahya B. M., and Seker D. Z. (2019). Designing Weather Forecasting Model Using Computational Intelligence Tools. Applied artificial intelligence, 33(2), 137-151.