Sabaragamuwa University Journal
Volume 20 Number 1 December 2022
ISSN 1391-3166 eISSN 2386-2041
http://dx.doi.org/10.4038/suslj.v20i1.7818

Rainfall forecasting model to make effective decisions on cultivation using multiple artificial neural networks – Case study Gampaha District

N.M. Hakmanage[1,2], N.V. Chandrasekara[1], and D.D.M. Jayasundara[1]

[1]Department of Statistics & Computer Science, Faculty of Science, University of Kelaniya, Sri Lanka.
[2]Department of Computer Systems Engineering, Faculty of Computing and Technology, University of Kelaniya, Sri Lanka.

*navodim@kln.ac.lk

Abstract

Rural communities in developing economies mainly depend on climate-sensitive activities such as agriculture for their livelihood and are particularly vulnerable to climate change, mainly to rainfall. Therefore, accurate forecasting of rainfall is vital for the sector of agriculture. This study aims at forecasting the rainfall for a farming village located in the Gampaha District, where the majority depends on agriculture. Artificial Neural Networks (ANN) can be identified as a prominent technique for forecasting among all forecasting techniques which was inspired by human brain neural systems. ANNs were employed in the study to build the rainfall forecasting model. The other climatic factors: temperature, humidity, pressure, wind speed, and cloud percentage were considered as affecting factors for rainfall in the model building process. Previously recorded data were collected from several stations near the study area. This study expanded to develop multiple artificial neural networks using Feed Forward Neural Networks (FFNN) for each climatic factor separately and used the forecasted values of those models as input to the rainfall forecasting model. Models were trained thousands of times by changing the model parameters and obtaining optimal parameters. The model was finalized based on the minimum mean squared error of 0.0547 and minimum normalized mean squared error of 0.378. With the involvement of many climatic factors, the proposed model leads to identify the rainfall variability which will be potentially allow farmers and others in the agricultural sector to make decisions on reducing unwanted impacts or take advantage of expected favorable climate.

Keywords: Artificial Neural Networks, rainfall, forecasting, climate-sensitive, cultivation

1.Introduction

Rainfall forecasting is critical for preventing flooding and managing water supplies, for the agricultural sector, as well as preserving lives and property and ensuring economic activity. Water supplies, water quality, and the aquatic ecology all suffer when there is not enough rain. As agriculture is climate sensitive, it is very important to recognize the climatic changes early in decision making of agriculture. Among those climatic factors, rainfall is one of the major influential climatic factors for cultivation (Hakmanage et al, 2021). Thus, this study aims to develop an accurate rainfall forecasting model which will be beneficial to farming communities.

The ability to predict rainfall several months ahead of time, if achievable, would allow for more efficient water usage. Several methods are available for rainfall forecasting, such as numerical weather prediction models, statistical methods, and machine learning techniques. Among these, machine learning techniques, such as artificial neural network (ANN), k-nearest neighbor, support vector machine, and random forest model, are more suitable for rainfall forecasting because physical processes affecting rainfall occurrence are highly complex and non-linear. The ANN is a form of machine learning technique that has been widely used in rainfall prediction given its ability to identify highly complex non-linear relationships between input and output variables without the need to understand the nature of the physical processes. The artificial neural network is a type of data driven technique which refers to the principle of biological neurons (Seyyedabbasi et al., 2018).

This was firstly introduced by McCulloch and Pitts in 1943 (McCulloch et al.,1943). Hu (1964) implemented the ADALINE system for weather forecasting, which became one of the earliest applications of ANN in weather forecasting. The most common neural network used in predictions in past studies is the feed forward neural network (Vamsidhar et al., 2010). Back propagation is a widely used algorithm for training feed forward neural networks Shaikh et al., 2017).

The low accuracy of long-term forecasts is a common issue with time series forecasting models (Seyyedabbasi et al., 2018). The estimated value of a variable can be relatively reliable in the short term, but the calculation is likely to become less accurate in the long run. There are a number of possible explanations for the growing inaccuracy. One reason is that the environment in which the model was developed has changed over time. As a result, the input that is true for a given time period has no bearing on the output that is meaningful for a time interval that is far in the future. Another reason is that the model itself was not well developed. Improper preparation or a lack of adequate data for training causes inaccuracy. The trained model may cover the immediate surroundings of the data set, but it fails to model cyclic changes in trend or seasonal trends. The third cause of inaccuracy is propagation errors due to a lack of future values in the input combination to model predictions. If it required values at time t in the input combination to predict time t step ahead, then we need to identify the error that occurs due to unavailability of data at time t. This error accumulates over time and can exceed a reasonable threshold for long-term forecasts. To address the third problem, this study proposes a multiple neural network model that combines short-term and long-term neural networks to accommodate a wide range of prediction terms. In this approach, several neural networks were built to predict climatic factors and they are combined into one model to forecast the rainfall. Hence, the objective of this research is to develop an Artificial Neural Network (ANN) model to forecast rainfall (mm) using other climatic factors such as temperature (°C), wind speed (kmph), air pressure (mb), humidity (%) and percentage of clouds by forecasting input combination individually using artificial neural networks. This study considered the records of climatic factors in the Dompe Divisional Secretariat in Gampaha District. The climatic data set includes 1560 observations of 12 variables including lags of the variables where k^{th} lag is the value at “k” time points before the time point t. First 80% of the data set which contained 1248 observations were used to train the network. Among these 12 variables temperature, wind speed, air pressure, humidity, and percentage of clouds were forecasted individually. Other inputs are lags of those variables. As it is required to present values of the variables in the input combination, this study used ANN to forecast those climatic factors: Temperature, Wind speed, Air pressure, Humidity, and Percentage of clouds. Finally, a multiple ANN model was developed to forecast rainfall with higher accuracy which was not found when revealing the literature. The novelty of this study is to redound the rainfall forecasting model using multiple ANNs. Multiple ANNs refer to a group of several ANNs which were performed separately and forecasted values of those models were used as inputs to the final model.

Western Province is a highly populated province in Sri Lanka with rapid urbanization with a land extent of 367,312 hectares. Gampaha District is one of the three Districts in the Western Province, which is urbanizing rapidly and cultivation still remains as a major livelihood in rural sectors of this district. The objective of this research is to analyze and forecast monthly rainfall using multiple artificial neural networks using related climatic factors: Temperature, Pressure, Humidity, Wind speed, and cloud percentage. Farmers will be informed in advance of the volume of rainfall through monthly rainfall forecasts. This information will aid them in determining what to cultivate and how much to cultivate during a given season.

2. Literature Review

Artificial neural networks (ANN) have been widely applied to modeling data as it is flexible and capable of nonlinear modeling (Nanayakkara et al., 2014). Shaikh et al. (2017) observed that ANN has a very high potential to be used to forecast rainfall and it has better accuracy than statistical and mathematical models. On the other hand, ANN can be used for predictions as well as has the capability of examining and determining the nonlinear behavior of historical data used for prediction (Kolarik and Rudorfer, 1994).

Several models of ANNs have been developed for precipitation prediction. Ramirez et al. (2005) used a multi-layer feed-forward perceptron (MLFP) neural network for daily precipitation prediction in the region of Sao Paolo State, Brazil. In their research, potential temperature, the vertical component of the wind, specific humidity, air temperature, perceptible water, relative vorticity, and moisture divergence flux were used as input data for the training stage of the model. The results from this ANN model were much better than those obtained by the linear regression models. Hung et al. (2009) used 4 years of hourly data from 75 rain gauge stations in Bangkok, Thailand, to develop an ANN model for rainfall forecasting. The results showed that a generalized feedforward ANN model with hyperbolic tangent transfer functions achieved the best generalization of rainfall. Nanda et al. (2013) used ARIMA (1,1,1) model and ANN models like multi-layer perceptron (MLP), functional-link artificial neural network (FLANN) and Legendre polynomial equation (LPE) to predict the rainfall time series data in India. MLP, FLANN, and LPE gave very accurate results for complex rainfall time series model with minimum absolute average percentage error (AAPE). Compared to ARIMA models, ANN models such as FLANN exhibit better prediction results with less AAPE for the analysis of rainfall time series in their study. Generally, there is no general ANN model defined as a suitable network that is able to work for all of the problems and for all different situations. Indeed, to find a proper ANN model, one should try to use different ANN models and structures to obtain an appropriate model that can work best for the particular situation of the case under study (Khalili 2006). However, according to the literature, FFNN networks have more reasonable outputs compared to other ANN types (Ninan et al. 2003; Ramirez et al. 2005; Kumar et al. 2012; Acharya et al. 2013). In this paper, ANNs have been used to obtain a prediction model for the monthly rainfall of selected study areas located in Gampaha district, Sri Lanka.

Hu (1964) initiated the implementation of ANN, an important soft computing methodology in weather forecasting. Since the last few decades, ANN, a voluminous development in the application field of ANN, has opened up new avenues to the forecasting task involving environment related phenomenon (Gardener and Dorling, 1998 and Bash et al., 2020). Michaelides et al (1995) compared the performance of ANN with multiple linear regressions in estimating missing rainfall data over Cyprus. Kalogirou et al (1997) implemented ANN to reconstruct the rainfall over the time series over Cyprus.Lee et al (1998) applied ANN in rainfall prediction by splitting the available data into homogenous subpopulations. Toth et al. (2000) compared short-time rainfall prediction models for real-time flood forecasting. Different structures of auto-regressive moving average (ARMA) models, ANN and nearest-neighbor approaches were applied for forecasting storm rainfall occurring in the Sieve River basin, Italy, in the period 1992- 1996 with lead times varying from 1 to 6 hours. The ANN adaptive calibration application proved to be stable for lead times longer than 3 hours, but inadequate for reproducing low rainfall events. Koizumi (1999) employed an ANN model using radar, satellite and weather-station data together with numerical products generated by the Japan Meteorological Agency (JMA) Asian Spectral Model and the model was trained using 1-year data. It was found that the ANN performed better than the persistence forecast (after 3 hours), the linear regression forecasts, and the numerical model precipitation prediction. As the ANN model was trained with only 1-year data, the results were limited. When more training data became available, the author believed that the neural networkś performance would improve. It is still unclear how much each predictor contributed to the forecast, and whether recent observations could help improve it.

3.Materials and Methods

Artificial neural networks were used to build forecasting models for temperature, air pressure, humidity, wind speed and cloud percentages separately. Forecasted values of each model were used as inputs to the rainfall forecasting model. Here, feed forward neural network models were trained with Levenberg Marquardt back propagation algorithm. Then a combination of inputs and components of the neural network was adjusted to obtain the most accurate model. Finally, model comparison was done to identify the best model for each data set based on the forecasting error.

3.1.Preliminary analysis

The feature selection was carried out under the preliminary analysis. The relationship between original variables and the cross correlations were assessed using proper statistical tests in order to identify significant variables for the forecasting models. The Person's correlation coefficient and the Spearman's correlation coefficient were considered in the correlation analysis to identify linear and non-linear relationships among variables respectively.

3.1.1.Person's correlation coefficient (r)

Person's correlation coefficient measures the linear association between two variables. In here best fit line drawn through data of two variables and r represent the distance between data points and this line. r can take values in between -1 and +1. The values greater than 0 indicate positive correlation and values less than 0 indicate negative correlation. If the value r is closer to 1 then there's a strong correlation between the two variables and if the value is closer to zero weak correlation is present.

3.1.2.Spearman's correlation coefficient (ρ)

The Spearman's rank-order correlation is the nonparametric version of the Pearson correlation coefficient. Spearman correlation can be used when assumptions in Pearson correlation are violated. In here spearman correlation can determine the strength and direction of monotonic relationship between paired data. The value of ρ lies between -1 and +1. The interpretation is similar to Pearson correlation.

3.2 Artificial Neural Network

The ANN is a computer program which is designed similar to the human brain where some cells are responsible for receiving the external stimulus, some for processing and some for the transfer of the response to the intended part. Neural network is made up of an input layer, one or more hidden layers and an output layer as shown in Fig. 1. All layers are interconnected by means of connectors of different weights. Similarities of human brain and ANN can be introduced as follows: Neuron - Cell/ Node, Synapse - Connections/Links, synaptic efficiency - weight/strength of the connection, firing frequency - output node. Hence, components of an ANN are inputs, outputs, weights, bias and activation functions. A multi-layer neural network contains more than one layer of artificial neurons or nodes. Different numbers of hidden neurons and different types of transfer functions can be used in layers (Shaikh et al., 2017). Multilayer neural networks are used in estimating nonlinear patterns. Feed forward neural network is a type of an artificial neural network where the connections between units do not form a cycle, that is information flows only in forward direction.

3.3 Training an Artificial Neural Network

To start the process of ANN, the initial weights are chosen randomly. Then, the training, or learning, begins. There are two approaches for training: - Supervised learning and Unsupervised learning. In supervised learning, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. In unsupervised training, the network is provided with inputs but not with desired outputs (Abhishek, 2012). Process of training ANN is the most important task in developing a neural network model. To obtain the best performance of ANN, following features will be changed one at a time: input combination, number of hidden layers, number of hidden neurons in each layer, transfer function in each layer, training algorithm, combination coefficient, increase/decrease factor of combination coefficient while keeping other factors constant. The Backpropagation algorithm is a supervised learning algorithm applied in multilayer feed-forward networks. The principle of the backpropagation approach is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised learning method, where the error between the system's output and a known expected output is presented to the system and used to modify its internal state. Technically, the backpropagation algorithm is a method for training the weights in a multilayer feed-forward neural network. There are several backpropagation algorithms available, including the Levenberg Marquardt back propagation algorithm, which updates weights and biases based on Levenberg Marquardt optimization. The Levenberg-Marquardt method uses the same approximation for the Hessian matrix, which is in

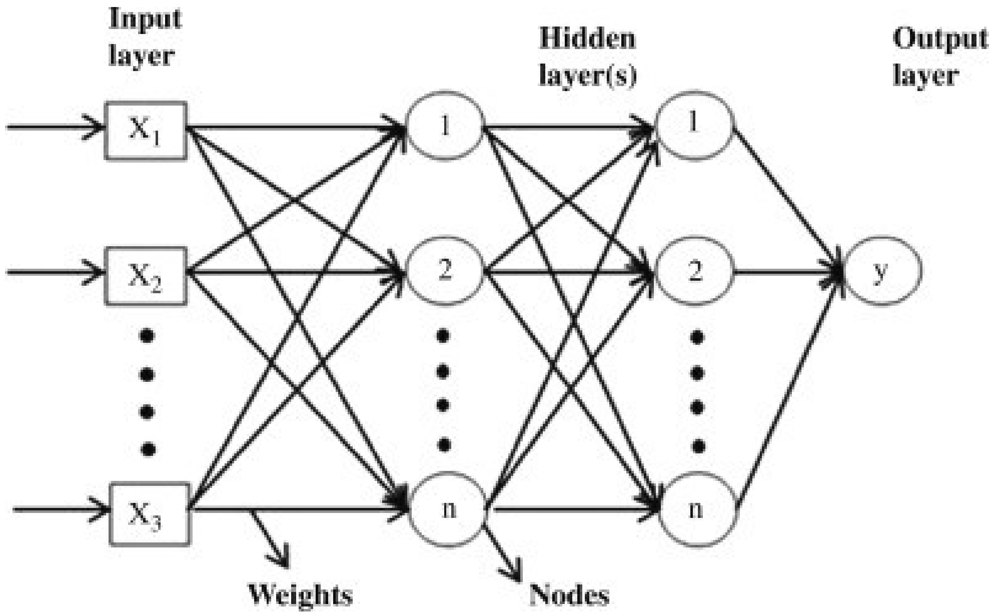

Figure 1: Multi-layer artificial neural network (ANN)

Eq. (1).

$$x_{\mathrm{k}+1} = x_{\mathrm{k}} x[J^{\mathrm{T}} J + \mu I] - 1 J^{\mathrm{T}} e \qquad (1)$$

where J is the Jacobean matrix which contains first derivatives of the network errors with respect to weights and biases. e is the vector of network errors. It is the identity matrix. μ is the combination coefficient. When μ becomes large the above equation approaches the gradient decent algorithm. When μ gets smaller it reaches the Gauss-Newton algorithm.

3.4 Data set

The data set contains monthly records of rainfall (mm), temperature (°C), wind speed (kmph), air pressure (mb), humidity (%) and percentage of clouds from January 2010 to December 2019 in Dompe divisional secretariat, Gampaha District. Data set was divided into three parts and used for training (80%), validation (10%) and testing (10%) before applying to each model. The training data set was used for the weight and bias updating process in each model. A practical way to find an optimal point of better generalization and avoid overfitting is to set aside a small percentage of the training data set named as validation data set. Finally, the performance of the models was evaluated using the test set.

3.5 Methodology

As rainfall is correlated with the temperature, wind speed, air pressure, humidity and percentage of clouds, those climatic factors were included in the input combination of the rainfall forecasting model. Even though it includes past values of those factors as lags, it also included the values at time t of other climatic factors to forecast values at time t+1. Hence long term prediction of rainfall using this model may cause error due to unavailability of future values of those inputs. Thus, this study conducted individual forecasting models for each climatic factor and used those forecasted values as inputs into the rainfall forecasting model. Structure of the proposed forecasting model using multiple ANN is given in Fig. 8.

3.6 Performance measures in forecasting models

The forecasting error was used to evaluate the performance of the proposed model. This is the difference between the forecasted and actual value. Mean Squared Error (MSE) is calculated using the Eq. (2).

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_{\mathrm{i}} - \hat{Y}_{\mathrm{i}})^2 \qquad (2)$$

Normalized Mean Squared Error (NMSE) is an estimator of overall deviation between forecasted value and actual value which is calculated using the Eq. (3).

$$NMSE = \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_{\mathrm{i}} - \hat{Y}_{\mathrm{i}})^2 \qquad (3)$$

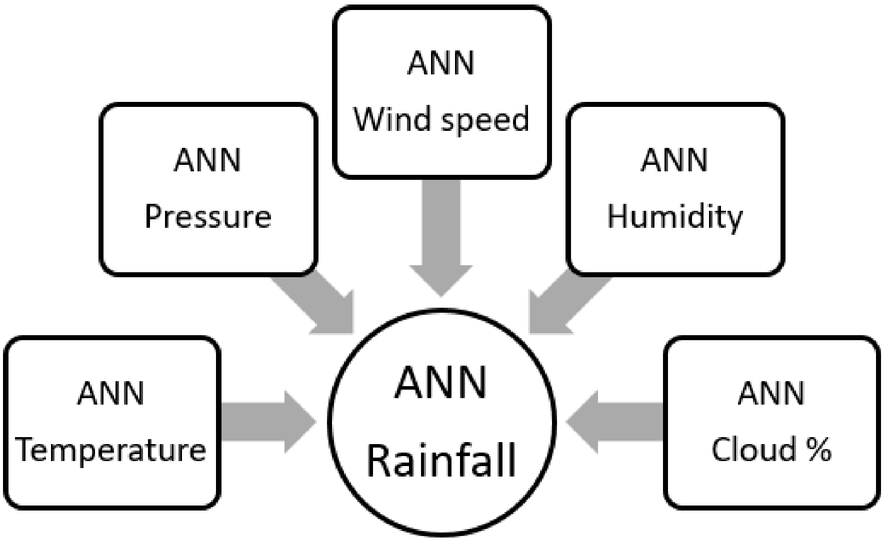

Figure 2: -Structure of the multiple ANN

4. Results and Discussion

This section describes relationships between variables and performance of the neural networks which were trained to forecast individual climatic factors separately in each trial under training process until reaching the optimal models. Further, the results of the final multiple ANN model are briefly summarized here.

4.1 Feature selection

In order to identify significant variables for forecasting models, the relationship between original variables and cross correlations was examined using suitable statistical tests. Table 1 represents the Person's and Spearman Rho correlations along with the p-value between rainfall and other selected climatic factors and cross correlations among them. These correlation coefficients were used to identify the input combination for the model as they are statistically significant (i.e. $p - value < 0.05$, all the significant relationships are given in Table 1).

4.2. Combination of input

The combination of inputs for each climatic factor was selected on the basis of the autocorrelation, partial autocorrelation, and moving averages (MA). Then, significant lags and MA were identified for each climatic factor. All identified features in the feature selection were considered as inputs to the ANNs and found the optimal input set using the performance of the test set. Table 2 illustrates the input combinations identified for each climatic factor along with the Mean Squared Error (MSE) and the Normalized Mean Squared Error (NMSE).

Input combinations of all climatic factors were limited to its first and second lags, which were obtained with minimum mean squared error and minimum normalized squared error.

4.3 Number of hidden layers

Number of hidden layers was changed from one to four for the models selected above and the results are recorded in Table 3.

There are two hidden layers in the ANN build for forecasting the temperature and humidity while one hidden layer in the ANN build for the forecasting the pressure, wind speed and cloud percentage which have minimum errors. Therefore, those models were used for further improvements.

4.4 Number of hidden neurons in the hidden layer

The data set was trained for several trials and observed the number of neurons in each hidden layer which captured the minimum error. As ANN models developed to forecast temperature and humidity, capturing two hidden layers, Table 4 (i) illustrates the number of neurons required for those models in each layer.

It was observed that 8 and 6 neurons in first and second hidden layers respectively in the ANN build for forecasting the temperature which has

Table 1: Correlation and cross-correlation between rainfall and other climatic factors

Parameter	Pearson's Correlation		Spearman Rho	
	Correlation	p-value	Correlation	p-value
Temperature (°C)	-0.114	0.001	-0.122	0.030
Wind speed (km/h)	-0.190	0.001	-0.194	0.001
Pressure (mB)	-0.279	0.001	-0.281	0.001
Humidity (%)	0.326	0.001	0.310	0.001
Cloud (%)	0.386	0.001	0.401	0.001
Temperature_lag1	0.248	0.001	0.243	0.001
Wind speed_lag1	-0.219	0.001	-0.229	0.001
Pressure_Lag1	-0.130	0.020	-0.130	0.007
Pressure_Lag2	-0.048	0.040	-0.056	0.030
Pressure_Lag3	-0.030	0.040	-0.083	0.030
Humidity_lag1	0.156	0.001	0.126	0.002
Cloud_Lag1	0.242	0.001	0.246	0.001

Table 2: Performance of models with different combinations of input variables

Variables	Input Combination	Model Performance	
		MSE	NMSE
Temperature	TM_lag1, TM_lag2	0.2257	0.2364
Pressure	PS_lag1, PS_lag2	0.2478	0.2130
Cloud percentage	CD_lag1, CD_lag2	0.2485	0.2369
Humidity	HM_lag1, HM_lag2	0.1457	0.1222
Wind speed	WS_lag1, WS_lag2	0.1987	0.1803

* Temperature (TM), Wind speed (WS), Pressure (PS), Cloud percentage (CD), Humidity (HD), Temperature$_{lag1}$ (TML1), Windspeed_lag1 (WSL1), Windspeed_lag2 (WSL2), Pressure_Lag1 (PSL1), Pressure_Lag2 (PSL2), Pressure_Lag3 (PSL3), Humidity_Lag1 (HDL1), Cloud_Lag1 (CDL1), L1-lag 1, L2-Lag 2, L3-Lag 3

Table 3: Performance of the models with different number of hidden layers

No. of hidden layers	Temperature		Pressure		Humidity		Wind speed		Cloud percentage	
	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE
1	0.2257	0.2364	**0.2478**	**0.2130**	0.1458	0.1221	**0.1987**	**0.1803**	**0.2485**	**0.2369**
2	**0.2121**	**0.2188**	0.2490	0.2568	**0.1246**	**0.1167**	0.2082	0.2071	0.2501	0.2544
3	0.3641	0.2789	0.2501	0.2544	0.2781	0.2498	0.2584	0.2203	0.2634	0.2578
4	0.3077	0.3047	0.2687	0.2614	0.3010	0.2765	0.2672	0.2644	0.3870	0.3109

minimum error. Further, based on the minimum MSE and NMSE it was identified that there are 12 neurons in both first and second layers in the ANN build for forecasting humidity.

There was one hidden layer in ANN models developed for forecasting the pressure, wind speed and the cloud percentage. Table 4 (ii) illustrates the number of neurons required for the hidden layer.

It was identified that there are 12, 8 and 7 neurons hidden layers in the ANN build for forecasting the pressure, wind speed and cloud percentage respectively based on the minimum error in each model.

Table 4 (i): Performance of the models with different number of hidden neurons under 1st and 2nd hidden layers

Number of Neurons		MSE	NMSE
Layer 1	Layer 2		
Temperature			
7	8	0.2780	0.3344
7	9	0.3881	0.3478
8	5	0.3748	0.3627
8	**6**	**0.2165**	**0.2142**
8	7	0.2740	0.2355
8	8	0.2783	0.3007
9	9	0.3578	0.3422
9	10	0.3047	0.3881
10	8	0.3742	0.3077
10	9	0.3904	0.3199
10	10	0.4369	0.4071
Humidity			
10	9	0.1245	0.1169
10	10	0.1378	0.1346
10	11	0.1480	0.1631
10	12	0.1495	0.1385
11	11	0.1663	0.1451
11	12	0.1879	0.1602
12	10	0.1387	0.1300
12	11	0.1268	0.1278
12	**12**	**0.1208**	**0.1143**
13	10	0.2047	0.2382
13	11	0.2525	0.2366
13	12	0.2877	0.2361
14	10	0.3781	0.3408

4.5 Transfer function in each layer

There are some transfer functions used in neural networks. Log-sigmoid, tan sigmoid and pure linear are the most popular functions among them. The neural network was trained several times to cover all possible combinations of transfer functions by changing transfer function in layers. Selected transfer functions for each layer are illustrated in Table 5 (i) and (ii) for each climatic factor.

It was identified that the transfer function of the Input hidden layer 1 as log-sigmoid, Hidden layer 1 to hidden layer 2 as log-sigmoid and Hidden layer 2 to output layer as pure linear function for the ANN model build for temperature while the transfer function of the Input hidden layer 1 as log-sigmoid, Hidden layer 1 to hidden layer 2 as tan-sigmoid and Hidden layer 2 to output layer as pure linear function for the ANN model for humidity depend on the minimum errors of the test set.

4.7 Change the parameters of the Algorithm

After selecting the training algorithm, the model was trained by changing the model parameter: combination coefficient (μ) until the minimum error was obtained. The default value of μ was 0.001 and it was changed from 0.00001 to 0.01 When the combination coefficient 0.0001 for Levenberg- Marquardt back-propagation algorithm, the ANN model exhibits the minimum error (Table 7). Therefore, suitable values of μ for models were decided as 0.0001 for the ANN forecasting models for temperature, pressure and wind speed and 0.0005 for cloud percentage ANN model and 0.005 for humidity forecasting ANN model.

The minimum errors yield the transfer functions for the Input hidden layer 1 and Hidden layer 1 to hidden layer 2 are log-sigmoid and tan-sigmoid, log-sigmoid and pure linear, tan sigmoid and log-sigmoid for the ANN models for pressure, wind speed, cloud percentage respectively.

4.6 Training algorithm

There are several backpropagation training algorithms available which are used to train the neural networks. Out of these training algorithms Levenberg-Marquardt back-propagation algorithm yielded the minimum NMS and NMSE for all models (Table 6). 'trainlm' function yields the best performance while the 'trainbr' function gives the next best performance. After several trials it was decided that 'trainlm' was the best algorithm which has the ability to converge faster.

4.8 Architecture of the feedforward neural network model

With those adjustments the model improving process finally five forecasting models were suggested for the forecasting of the climatic factors: temperature, pressure, humidity, wind speed and cloud percentage using the minimum errors as highlighted in Table 7. Architectures of those neural networks are given in Fig. 3.

Forecasted values and the actual values of the test set were plotted in Fig. 4 together to visualize the performance of the finally identified models in forecasting the climatic factors: temperature, pres-

Table 4 (ii): Performance of models with different numbers of hidden neurons under one hidden layer

No. of Neurons in Layer	Pressure		Wind speed		Cloud percentage	
	MSE	NMSE	MSE	NMSE	MSE	NMSE
1	0.4369	0.4071	0.4028	0.3907	0.2485	0.2369
2	0.3047	0.3081	0.3999	0.3987	0.2301	0.2270
3	0.3748	0.3627	0.3666	0.3870	0.2280	0.2207
4	0.3904	0.3899	0.3904	0.3887	0.2358	0.2135
5	0.3858	0.3477	0.2908	0.2807	0.2178	0.2038
6	0.3783	0.3307	0.2783	0.2307	0.1938	0.1759
7	0.3578	0.3422	0.1925	0.1878	**0.1578**	**0.1422**
8	0.2778	0.2530	**0.1723**	**0.1603**	0.1778	0.1597
9	0.2863	0.2599	0.1987	0.1803	0.2163	0.2099
10	0.2301	0.2200	0.2047	0.2081	0.2201	0.2200
11	0.2280	0.2207	0.2748	0.2627	0.2345	0.2207
12	**0.2208**	**0.2103**	0.2904	0.2899	0.2423	0.2178
13	0.2347	0.2309	0.3858	0.3477	0.3904	0.3890
14	0.2351	0.2343	0.3783	0.3307	0.3858	0.3478
15	0.2423	0.2402	0.4280	0.4027	0.3583	0.3336

Table 5 (i): Performance of models with different transfer functions in each layer

Input to hidden layer 1	Hidden layer 1 to hidden layer 2	Hidden layer 2 to output layer	Temperature		Humidity	
			MSE	NMSE	MSE	NMSE
Log-sigmoid	Log-sigmoid	Log-sigmoid	0.2374	0.2345	0.1208	0.1142
Log-sigmoid	Log-sigmoid	Pure linear	**0.2143**	**0.2113**	0.1568	0.1332
Pure linear	Log-sigmoid	Tan-sigmoid	0.2987	0.3014	0.2089	0.1987
Log-sigmoid	Tan-sigmoid	Pure linear	0.3780	0.3105	**0.1187**	**0.1106**
Pure linear	Tan-sigmoid	Tan-sigmoid	0.3328	0.3780	0.1356	0.1278
Tan-sigmoid	Tan-sigmoid	Log-sigmoid	0.3078	0.3792	0.1236	0.1158

* Values in bold indicate the best performance for the respective variable.

Table 5 (ii): Performance of models with different transfer functions in each layer

Input to hidden layer 1	Hidden layer 1 to output layer	Pressure		Wind speed		Cloud percentage	
		MSE	NMSE	MSE	NMSE	MSE	NMSE
Log-sigmoid	Log-sigmoid	0.2208	0.2103	0.1880	0.1628	0.1578	0.1422
Log-sigmoid	Pure linear	0.2531	0.2178	**0.1723**	**0.1603**	0.1367	0.1178
Log-sigmoid	Tan-sigmoid	**0.2147**	**0.2023**	0.1971	0.1874	0.1369	0.1123
Tan-sigmoid	Tan-sigmoid	0.2387	0.2246	0.2144	0.2036	0.1378	0.1230
Tan-sigmoid	Log-sigmoid	0.2439	0.2371	0.2470	0.2469	**0.1056**	**0.1023**
Tan-sigmoid	Pure linear	0.2498	0.2475	0.2698	0.2580	0.1155	0.1369
Pure linear	Pure linear	0.2457	0.2498	0.3066	0.2907	0.2151	0.2036
Pure linear	Tan-sigmoid	0.2689	0.2571	0.3874	0.3571	0.2378	0.2169
Pure linear	Log-sigmoid	0.2678	0.2419	0.3706	0.3412	0.2588	0.2274

* Values in bold indicate the best performance for the respective variable.

Table 6: Performance of models with different backpropagation training algorithms.

Back propagation algorithm	Temperature		Pressure		Humidity		Wind speed		Cloud percentage	
	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE
trainlm	**0.2143**	**0.2113**	**0.2147**	**0.2023**	**0.1187**	**0.1106**	**0.1723**	**0.1603**	**0.1056**	**0.1023**
trainbr	0.2512	0.2503	0.2389	0.2158	0.1220	0.1208	0.1839	0.1733	0.1189	0.1366
trainbfg	0.3422	0.3780	0.2701	0.2863	0.2781	0.2369	0.2722	0.2389	0.1654	0.1354
trainrp	0.4038	0.4218	0.2978	0.2874	0.3038	0.2218	0.4080	0.4436	0.2362	0.2249
trainscg	0.3018	0.3746	0.2903	0.2873	0.3023	0.2987	0.4128	0.4632	0.2477	0.2301
traincgb	0.4281	0.4777	0.3288	0.3182	0.3281	0.3177	0.6038	0.5702	0.2698	0.2182
traincgf	0.3587	0.4132	0.3975	0.3214	0.3588	0.3132	0.4389	0.3708	0.3981	0.3236
traincgp	0.4586	0.4889	0.4080	0.4436	0.3680	0.3481	0.3358	0.3731	0.2080	0.1360
trainoss	0.5236	0.5111	0.4128	0.4632	0.3872	0.3462	0.5077	0.4618	0.2128	0.1632
traingdx	0.5478	0.5781	0.4955	0.4287	0.4369	0.4125	0.4189	0.4122	0.3555	0.3207
traingdm	0.5681	0.5147	0.5238	0.5147	0.4415	0.4114	0.6039	0.5477	0.3688	0.3111
traingd	0.6877	0.6238	0.5511	0.5102	0.4876	0.4025	0.5397	0.5333	0.4587	0.3022

* Values in bold indicate the best performance for the respective variable.

Table 7: Performance of models with different combination coefficients

Combination coefficient μ	Temperature		Pressure		Humidity		Wind speed		Cloud percentage	
	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE	MSE	NMSE
0.00001	0.2143	0.2113	0.2378	0.2017	0.2236	0.2138	0.2878	0.2278	0.2933	0.2071
0.00005	0.2047	0.2135	0.2136	0.1789	0.2174	0.2036	0.1736	0.1789	0.1368	0.1274
0.0001	**0.1036**	**0.1002**	**0.1172**	**0.1023**	0.1975	0.1748	**0.1674**	**0.1520**	0.1103	0.1179
0.0005	0.1558	0.1358	0.1238	0.1311	0.1548	0.1357	0.1689	0.1587	**0.1014**	**0.1011**
0.001	0.1378	0.1477	0.2147	0.2023	0.1187	0.1146	0.1723	0.1603	0.1056	0.1023
0.005	0.1890	0.1972	0.1587	0.1508	**0.1136**	**0.1072**	0.1890	0.1823	0.1123	0.1170
0.01	0.2078	0.2041	0.1996	0.1278	0.2511	0.2236	0.1990	0.1987	0.1190	0.1136

* Values in bold indicate the best performance for the respective variable.

Table 8: Performance of models with different combinations of input variables

Input Variable Combination	Model Performance	
	MSE	NMSE
TM, WS, PS, HD, CD, TL1, WS1, WSL2, PSL1, PSL2, PSL3, HDL1, CDL1	0.2504	0.2689
TM, WS, PS, HD, CD, TL1, WS1, PSL1, PSL2, PSL3, HDL1, CDL1	**0.2444**	**0.2158**
TM, WS, PS, HD, CD, TL1, WS1, PSL1, PSL3, HDL1, CDL1	0.2756	0.3541
TM, WS, PS, HD, CD, TL1, WS1, PSL1, HDL1, CDL1	0.2950	0.2975
WS, PS, HD, CD, TL1, WS1, PSL1, HDL1, CDL1	0.3901	0.4986
WS, PS, HD, CD, TL1, WS1, HDL1, CDL1	0.3849	0.3120
WS, PS, HD, CD, TL1, WS1, CDL1	0.3834	0.4739
WS, PS, HD, CD, TL1, WS1, WSL2, PSL1, PSL2, PSL3, CDL1	0.3378	0.5438
TM, WS, PS, HD, CD, TL1, WS1, CDL1	0.3380	0.4121

* The input variable combinations represent different sets of features used in the model, where TM = Temperature Mean, WS = Wind Speed, PS = Pressure, HD = Humidity Daily, CD = Cloud Density, TL1 = Temperature Lag 1, WS1 = Wind Speed Lag 1, WSL2 = Wind Speed Lag 2, PSL1 = Pressure Lag 1, PSL2 = Pressure Lag 2, PSL3 = Pressure Lag 3, HDL1 = Humidity Daily Lag 1, CDL1 = Cloud Density Lag 1. Bolded values indicate the best-performing combination based on MSE and NMSE.

(A)

(B)

(C)

(D)

(E)

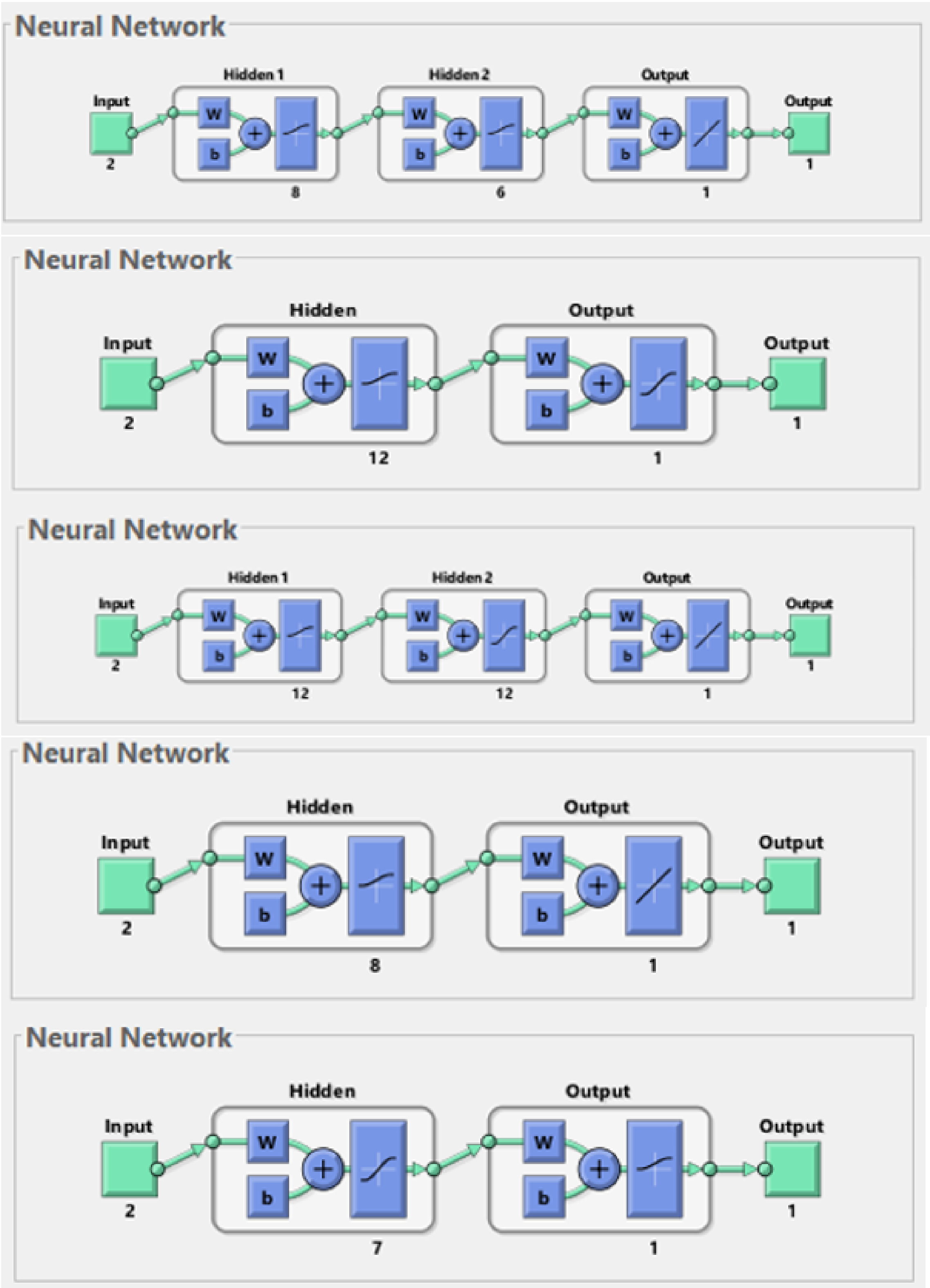

Figure 3: Architecture of the feedforward neural network model (A) Temperature, (B) Pressure, (C) Humidity, (D) Wind speed and (E) cloud percentage

sure, humidity, wind speed and cloud percentage. According to the Fig. 4, it can be identified that the final feed forward neural network models are capable of forecasting given climatic factors with higher accuracy.

Table 9: Performance of models with different numbers of neurons in the hidden layer

Number of Neurons in Hidden Layer	Model Performance	
	MSE	NMSE
1	**0.2444**	0.2689
2	0.2862	**0.2174**
3	0.3201	0.2541
4	0.3440	0.3957

Table 10: Performance of models with different numbers of neurons in a single hidden layer

Number of Neurons in Hidden Layer	Model Performance	
	MSE	NMSE
1	0.6447	0.6890
2	0.5298	0.5888
3	0.5127	0.5114
4	0.4389	0.5301
5	0.4011	0.5078
6	0.4712	0.5199
7	0.6882	1.6071
8	0.3784	0.6782
10	0.5299	0.5972
11	0.5972	0.8714
12	0.3678	0.3538
13	**0.3001**	**0.2927**
14	0.4102	0.3927
15	0.5245	0.3756
16	0.5189	0.4893
17	0.4156	0.6721
18	0.4718	0.5748
19	0.5921	1.3687
20	0.8670	2.5911

4.9 Rainfall forecasting model

According to the results of Table 1, there are 12 variables in the data set which are significantly correlated with rainfall and those variables are used as inputs to the rainfall forecasting model. There are several possible combinations of 12 input variables which can be identified using $^{n}C_{r}$ calculation. In practice it is difficult to test all these combinations one by one. Thus this study was carried out using most important combinations based on correlation coefficients which indicate the relationship between variables. Then variables were removed one by one and observed the MSE and NMSE as given below.

Table 11: Performance of models with different numbers of neurons in each of two hidden layers

Number of Neurons		Model Performance	
Layer 1	Layer 2	MSE	NMSE
10	11	0.4145	0.5145
10	12	0.6675	1.5961
10	14	0.3645	0.5712
11	12	0.6109	0.6098
12	10	0.5227	0.8385
12	11	0.3599	0.3387
13	9	0.4578	0.3785
13	**10**	**0.2967**	**0.2563**
13	11	0.7458	0.3482
13	12	0.7420	0.6998
14	10	0.4875	0.5207
14	11	0.4120	0.4868
14	12	0.5844	1.2998
15	12	0.8536	2.5782

According to the results in Table 8, the ANN model with a combination with 12 inputs exhibits the minimum error. Thus this input set was used for further improvements of the model.

Number of hidden layers was changed from one to four for the model selected above and the performance of the ANN models was recorded in Table 9. According to the results, minimum NMSE is recorded under two hidden layers while minimum MSE was recorded under one hidden layer. Thus, it was decided to proceed the training of the neural network with both one and two hidden layers separately.

For a single hidden layer, number of neurons was identified by changing the number of neurons in the hidden layer from 1 to 20. Then considered two hidden layers and the most suitable number of neurons in each hidden layer were identified by changing the number of neurons in the first hidden layer from 1 to 20 while changing the number of neurons in the second layer from 1 to 20 simultaneously.

Train the data set for several trials and observed that the ANN model with 13 neurons in the hidden layer displayed the minimum error. Then trained and observed the results of the neural network with two hidden layers (Table 10).

Train the data set for several trials and observed that the ANN model with 13 neurons in the hidden

Table 12: Performance of models with different transfer functions in each layer

Transfer Functions			Model Performance	
Input to Hidden 1	**Hidden 1 to Hidden 2**	**Hidden 2 to Output**	**MSE**	**NMSE**
Log-sigmoid	Log-sigmoid	Log-sigmoid	0.5372	0.3417
Pure linear	Log-sigmoid	Log-sigmoid	1.5879	0.4790
Log-sigmoid	Tan sigmoid	Log-sigmoid	0.8705	0.8702
Tan sigmoid	Pure linear	Log-sigmoid	0.6423	0.7410
Tan sigmoid	Log-sigmoid	Log-sigmoid	0.9457	0.9712
Pure linear	Pure linear	Log-sigmoid	3.2470	4.2801
Log-sigmoid	Pure linear	Log-sigmoid	2.5834	0.9742
Tan sigmoid	Tan sigmoid	Log-sigmoid	2.1470	3.0271
Log-sigmoid	**Log-sigmoid**	**Pure linear**	**0.2374**	**0.2272**
Pure linear	Log-sigmoid	Pure linear	0.3455	0.3852
Log-sigmoid	Tan sigmoid	Pure linear	0.7205	0.9692
Tan sigmoid	Pure linear	Pure linear	0.7899	0.8454
Tan sigmoid	Log-sigmoid	Pure linear	5.3578	0.9712
Pure linear	Pure linear	Pure linear	7.1287	5.8702
Log-sigmoid	Pure linear	Pure linear	3.5874	2.8701
Tan sigmoid	Tan sigmoid	Pure linear	4.2876	5.2389
Log-sigmoid	Log-sigmoid	Tan sigmoid	0.6188	0.4279
Pure linear	Log-sigmoid	Tan sigmoid	2.6890	0.6387
Log-sigmoid	Tan sigmoid	Tan sigmoid	0.8678	0.9025
Tan sigmoid	Pure linear	Tan sigmoid	0.9155	0.6328
Tan sigmoid	Log-sigmoid	Tan sigmoid	0.8970	0.7458
Pure linear	Pure linear	Tan sigmoid	2.4777	1.0472
Log-sigmoid	Pure linear	Tan sigmoid	4.2891	0.9742
Tan sigmoid	Tan sigmoid	Tan sigmoid	4.1880	1.7781

Table 13: Performance of models with different backpropagation training algorithms

Training Algorithm	Model Performance	
	MSE	**NMSE**
Levenberg-Marquardt back-propagation	**0.0975**	**0.2058**
Bayesian Regularization back-propagation	0.2967	0.2563
BFGS Quasi-Newton back-propagation	0.4822	0.6572
Resilient Back propagation	2.0070	2.4948
Scaled Conjugate Gradient back-propagation	0.6954	0.8405
Conjugate Gradient with Powell/Beale Restarts back-propagation	0.6315	1.8077
Fletcher-Powell Conjugate Gradient back-propagation	0.5340	1.8725
Polak-Ribiere Conjugate Gradient back-propagation	0.6871	1.2224
One Step Secant back-propagation	0.7186	1.8494
Variable Learning Rate Gradient Descent back-propagation	0.6572	0.8430
Gradient Descent with Momentum back-propagation	2.2034	3.4780
Gradient Descent back-propagation	1.1772	2.8145

layer displayed the minimum error. Then trained and observed the results of the neural network with two hidden layers (Table 10).

After training the data set for multiple trials,

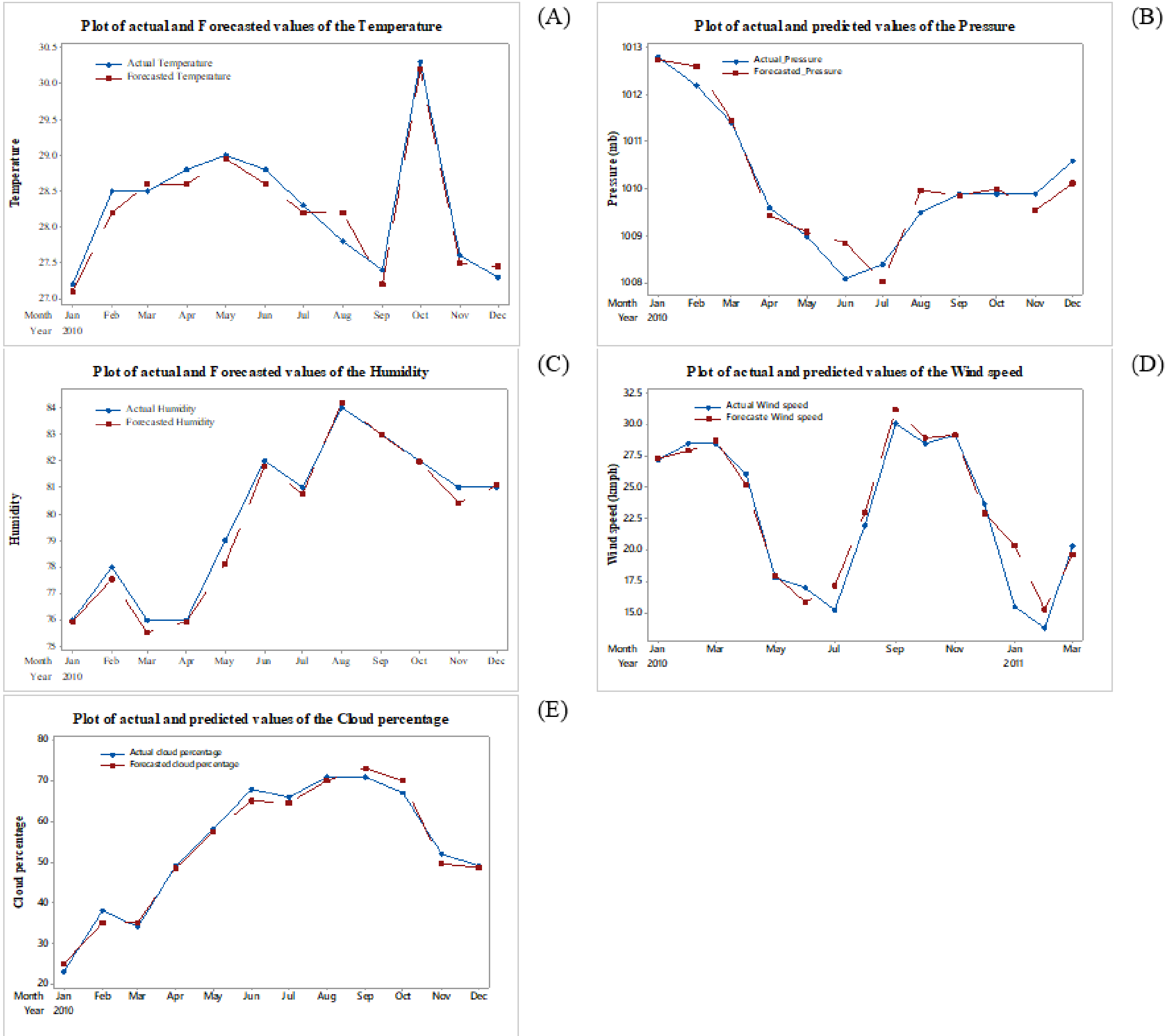

Figure 4: Plot of actual and predicted values of the climatic factors for the test set of (A) Temperature, (B) Pressure, (C) Humidity, (D) Wind speed and (E) cloud percentage

it was discovered that the minimum error was captured by 13 neurons in the first hidden layer and 10 neurons in the second hidden layer. As a result, the neural network will be trained using 13 and 10 neurons in the first and second hidden layers, respectively. (Table 11).

There are many transfer functions used in neural networks. Log-sigmoid, tan sigmoid and pure linear are the most popular functions among them. It was identified that the transfer function log-sigmoid is suitable for hidden layers one and two while a pure linear function is appropriate for the output layer (Table 12).

Your paragraph text goes here. This discusses the transfer function analysis and leads into the results shown in Table 12.

The next paragraph continues here...

There are several backpropagation training algorithms available which are used to train the neural networks. According to table 13, out of these training algorithms the model with the Levenberg-Marquardt back-propagation algorithm yields the minimum NMS and NMSE.

After selecting the training algorithm, trained the model by changing the model parameter: combination coefficient (μ) until it gets the minimum error where the default value is 0.001. In Here value of μ was changed from 0.00001 to

0.01 and observed the change in NMSE of the test data set as shown in Table 14.

Table 14: Performance of models with different backpropagation learning rates

Learning Rate μ	Model Performance	
	MSE	**NMSE**
0.00001	0.1282	0.3784
0.00005	0.0998	0.2758
0.0001	**0.0547**	**0.0378**
0.0005	0.0997	0.2874
0.001	0.0975	0.2058
0.005	0.1050	0.2148
0.01	0.2410	0.5782

The combination coefficient 0.0001 for Levenberg-Marquardt back-propagation algorithm yields the minimum error. Therefore, the suitable value of μ for this model was finalized as 0.0001.

With those adjustments the model improving process completed and finalized the multiple feed forward neural network model to forecast rainfall. The architecture of the final ANN model is given in Fig. 5.

Different networks give different results with the same training functions and adaptive learning functions having the same number of neurons. Therefore, thousands of models were trained until they got the minimum error in each forecasting model. The model developed to forecast rainfall requires climatic records of current month to forecast rainfall for next month and future. Thus, this study focuses on obtain the current climatic conditions using individual climatic factors forecasting models. Hence, initially following forecasting models were developed.

Temperature forecasting model includes 2 variables: TM_lag1 and TM_lag2. There are two hidden layers with 8 and 6 hidden neurons in layer 1 and 2 respectively. Log sigmoid, log sigmoid and pure linear transfer functions are used in hidden layer 1, 2 and output layer respectively. The network is trained with the 'trainlm' function. The optimal combination of coefficient is 0.0001 with 0.04 as the decreasing factor and 25 as the increasing factor.

PS_lag1 and PS_lag2 are two variables in the final model developed for pressure forecasting. In the hidden layer, there is one hidden layer with 12 hidden neurons. In the hidden layer 1 and output layer, respectively, log sigmoid and tan sigmoid transfer functions were used. The 'trainlm' function is used to train the network. The optimum coefficient combination is 0.0001 with a decreasing factor of 0.04 and an increasing factor of 25.

Based on the NMSE of the test data set, the optimal model for forecasting humidity was selected after thousands of trials. HM_Lag1 and HM_lag2 are two variables in the final model. In each of the two hidden layers, there are 12 hidden neurons in layer 1 and 12 hidden neurons in layer 2. Hidden layer 1, 2, and output layer utilize log sigmoid, tan sigmoid, and pure linear transfer functions, respectively. The 'trainlm' function is used to train the network. The optimum coefficient combination is 0.005 with a decreasing factor of 0.04 and an increasing factor of 25.

WS_lag1 and WS_lag2 are two variables in the model designed to forecast wind speed. In the hidden layer, there is one hidden layer with eight hidden neurons. In the hidden layer 1 and output layer, log sigmoid and pure linear transfer functions were employed, respectively. The 'trainlm' function is used to train the network. The optimum coefficient combination is 0.0001 with a decreasing factor of 0.04 and an increasing factor of 25.

The final model, which comprises two variables: CD lag1 and CD lag2, is used to forecast cloud percentage. In the hidden layer, there is one hidden layer with seven hidden neurons. In the hidden layer 1 and output layer, the tan sigmoid and log sigmoid transfer functions are employed, respectively. The 'trainlm' function is used to train the network. The optimum coefficient combination is 0.0005 with a decreasing factor of 0.04 and an increasing factor of 25.

Then, forecasted values of the above mentioned ANN models are inputs to the rainfall forecasting model as that model requires climatic records of current month to forecast rainfall for next month and future. The rainfall forecasting model includes 12 variables: temperature, wind speed, pressure, cloud percentage, humidity, and their derivatives such as temparature_lag1, windspeed_lag1, pressure_lag1, pressure_lag2, pressure_lag3, humidity_lag1, and cloud_lag1. There are two hidden layer with 13 and 10 hidden neurons in the first and second layer respectively. Log sigmoid transfer

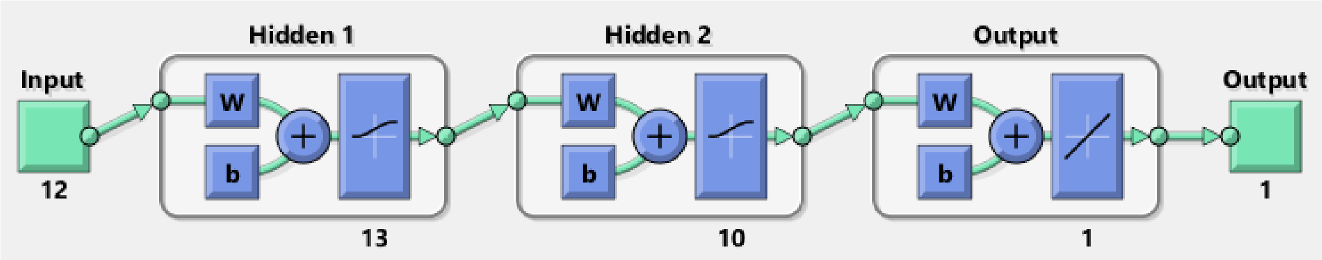

Figure 5: Architecture of the Feedforward Neural Network model for rainfall forecasting

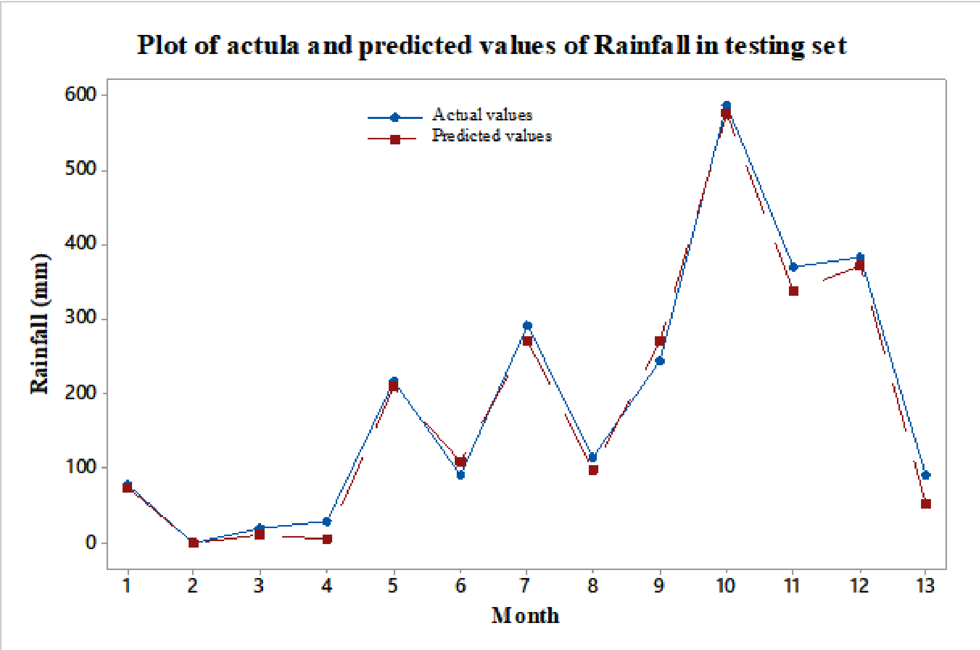

Figure 6: Plot of actual and predicted values of the rainfall for the test set

function used in hidden layers 1 and 2, pure linear transfer function was used in the output layer as transfer functions. The network was trained with the Levenberg-Marquardt back-propagation algorithm with combination coefficient of 0.0001.

The test of the actual rainfall values and forecasted rainfall values were extracted and visualized in Fig.6.

Precision agriculture's ultimate goal is to maximize growth efficiency at the seed and plant level. While farmers must make several daily decisions based on climatic conditions, there are four main aspects of farming that are fundamentally impacted by weather such as crop growth, fertilizer timing and delivery, disease control and field workability. Therefore, the proposed multiple neural network model by this study will be an effective approach for farmers in the study area in decision making and helping them to plan efficiently, minimize costs and maximize yields. Multiple artificial neural network model is a new method in forecasting which contributes to overcome issues in predicting data inside a forecasting model.

Conclusion

A main objective of agricultural extension is to assist farmers in making decisions that will allow them to achieve their own objectives as efficiently as possible. Early determination of rainfall will lead to effective decision-making in cultivation. Hence, based on the minimum MSE and NMSE, rainfall forecasting model was developed with multiple ANNs which is important to farming communities in the selected study area.

- Temperature forecasting model: FFNN model forecast temperature with 0.1036 MSE and 0.1002 NMSE.
- Pressure forecasting model: FFNN model

capable of forecasting pressure with a MSE of 0.1172 and NMSE of 0.1023.

- Humidity forecasting model: FFNN model was able to forecast humidity with 0.1136 MSE and 0.1072 NMSE.
- Wind speed forecasting model: FFNN network was capable of forecasting wind speed with 0.1674 MSE and 0.1520 NMSE.
- Cloud percentage forecasting model: FFNN model forecast the cloud percentage with 0.1014 MSE and 0.1011 NMSE.

Forecasted values of the above mentioned models were considered as inputs to the model which is developed to forecast rainfall. The multiple ANN rainfall forecasting model introduced in this study is capable for forecasting rainfall with higher accuracy which exhibits MSE of 0.0547 and NMSE of 0.378. Thus, it will be a benefit to the farming communities to access the rainfall variations early and make more effective decisions on cultivation. This study transfers to the farming communities via a user-friendly web application or mobile application.

Acknowledgment

This research was financed by the Accelerating Higher Education Expansion and Development (AHEAD) Development Oriented Research (DOR) grant of the Ministry of Higher Education funded by the World Bank.

References

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*, pages 82–87. IEEE.

Basha, C. Z., Bhavana, N., Bhavya, P., and Sowmya, V. (2020). Rainfall prediction using machine learning & deep learning techniques. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 92–97. IEEE.

Hakmanage, N. M., Chandrasekara, N. V., and Jayasundara, D. D. M. (2021). Early Recognition of Rainfall to Plan and Manage the Cultivation: A Forecasting Model Using Artificial Neural Networks for Dompe Divisional Secretariat in Gampaha District, Sri Lanka. *Tropical Agricultural Research*, 32(2):191–200.

Hu, M.J.C. (1964). *Application of ADALINE System to Weather Forecasting*. Department of Electrical Engineering, Stanford University.

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 & Earth System Sciences*, 13(8):1413–1425.

Khalili, N. (2006). *Rainfall prediction using artificial neural networks*. Master's thesis, Ferdowsi University of Mashhad, Iran.

Koizumi, K. (1999). An objective method to modify numerical model forecasts with newly given weather data using an artificial neural network. *Weather and Forecasting*, 14:109–118.

Kolarik, T. and Rudorfer, G. (1994). Time series forecasting using neural networks. *ACM SIGAPL APL Quote Quad*, 25(1):86–94.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Nanayakkara, K.A.D.S.A., Chandrasekara, N.V., and Jayasundara, D.D.M. (2014). Forecasting Exchange Rates using Time Series and Neural Network Approaches. *European International Journal of Science and Technology*, 3(2):65–73.

Ramirez, M. C. V., Ferreira, N. J., and Velho, H. F. (2005). Artificial neural network technique for rainfall forecasting applied to the Sao Paulo region. *Journal of Hydrology*, 301:146–162.

Seyyedabbasi, A., Candan, F., and Kiani, F. A. (2018). Method for forecasting weather condition by using artificial neural network algorithm. In *2018 2nd International Conference on e-Education, e-Business and e-Technology (ICEBT)*, pages 126–130. IEEE.

Shaikh, L. and Sawlani, K. (2017). A rainfall prediction model using artificial neural network. *International Journal of Technical Research and Applications*, 5(2):45–48.

Singh, P. and Borah, B. (2013). Indian summer monsoon rainfall prediction using artificial neural network. *Stochastic Environmental Research and Risk Assessment*, 27:1585–1599.

Sumi, S. M., Zaman, M. F., and Hirose, H. (2012). A rainfall forecasting method using machine learning models and its application to the Fukuoka city case. *International Journal of Applied Mathematics and Computer Science*, 22(4):841–854.

Toth, E., Brath, A., and Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology*, 239:132–147.

Vamsidhar, E., Varma, K. V. S. R. P., Rao, P. S., and Satapati, R. (2010). Prediction of rainfall using back propagation neural network model. *International journal on computer science and Engineering*, 2(4):1119–1121.

Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62.