

Artificial Neural Network Ensembles in Time Series Forecasting: an Application of Rainfall Forecasting in Sri Lanka

Harshani R. K. Nagahamulla, Uditha R. Ratnayake, Asanga Ratnaweera

Abstract— Weather forecasting is a widely researched area in time series forecasting due to the necessity of accurate weather forecasts in various human activities. Out of numerous weather forecasting techniques Artificial Neural Networks (ANN) methodology is one of the most widely used techniques. In this study the application of Neural Network Ensembles in Rainfall Forecasting is investigated by using various types of Ensemble Neural Networks (ENN) to forecast the rainfall in Colombo, Sri Lanka. Ensembles are generated by changing the network architecture, changing initial weights of the ANN and changing the ANN type. Two ensembles one consisting of a collection of networks of various architectures of Multi Layer Feed Forward Network with Back Propagation Algorithm (BPN), and the other consisting of a combination of BPN, Radial Basis Function Network (RBFN) and General Regression Neural Network (GRNN). The performance of ensembles are compared with the performance of BPN, RBFN and GRNN. The ANNs are trained, validated and tested using daily observed weather data for 41 years. The results of our experiment show that the performance of the ensemble models are better than the performance of the other models for this application and that changing the network type gives better results than changing the architecture of the ANN.

Index Terms— Rainfall forecasting, Artificial Neural Networks (ANN), Multi Layer Feed Forward Network (MLFFN), Back Propagation Algorithm (BPN), Radial Basis Function Network (RBFN), General Regression Neural Network (GRNN), Ensemble Neural Network (ENN)

I. INTRODUCTION

Weather forecasting is predicting the state of the atmosphere for a certain location for a certain time period. Data on previous and current atmospheric conditions and various scientific methods are used for this. Rainfall is the liquid form of precipitation. Quantitative precipitation forecasts are very important for planning day to day human

activities. Also for an agricultural based country like Sri Lanka long term plans and management activities like flood management, water resource management and agricultural planning activities depend on accurate rainfall predictions.

Ancient people observed weather patterns and predicted weather according to past weather occurrences. Nowadays there are many approaches to weather forecasting. Mathematical modelling, statistical modelling and artificial intelligence techniques are some of them. Using mathematical models of the atmosphere to predict future weather based on current weather conditions is called numerical weather prediction. This needs full knowledge of atmospheric dynamics and involves calculations with a large number of variables and huge datasets. Although this process requires a lot of computational resources due to the advancement of modern computer hardware there have been many improvements in numerical weather prediction [1]. Still there are difficulties in short term weather predictions because of sudden atmospheric changes.

Statistical weather forecasting methods mainly use time series analysis, time series forecasting and regression analysis using statistical models like autoregressive integrated moving average (ARIMA). These methods give excellent results in forecasting pressure and temperature but faces difficulties in forecasting precipitation accurately because the distribution of precipitation is bounded and skewed [2]. Nowadays a lot of researches are conducted on the use of intelligent techniques in rainfall forecasting. Genetic algorithms, fuzzy systems and ANN are some of them.

ANN is a forecasting tool that can handle complicated data efficiently. Different types of ANN exhibit different advantages. A collection of a finite number of ANN trained for the same task is called an ensemble neural network. Hansen and Salamon [3] explains that the generalisation ability of an ANN can be significantly improved through an ENN and the advantages of separate ANN can be combined to give a better result. In an ensemble separate ANN are trained individually and then their outputs are combined. The objective of this study is to investigate the appropriateness of ENN in rainfall forecasting and compare different methods of ensemble generating techniques. The performance of ENN is compared with BPN, RBFN and GRNN models.

The rest of this paper is organised as follows. Section II reviews the usage of ANN and ensembles in forecasting. Section III describes the ANN methodology, BPN, RBFN, GRNN and the ensemble techniques and a description of our

Harshani R. K. Nagahamulla is with the Dept of Computing & Information Systems, Faculty of Applied Sciences, Wayamba University of Sri Lanka, Kuliyaipitiya, Sri Lanka (e-mail: harshaninag@yahoo.com)

Uditha R. Ratnayake is with the Dept of Civil Engineering, Faculty of Engineering, University of Peradeniya Peradeniya, Sri Lanka (e-mail: udithar@pdn.ac.lk)

Asanga Ratnaweera is with the Dept of Mechanical Engineering, Faculty of Engineering, University of Peradeniya Peradeniya, Sri Lanka (e-mail: asangar@pdn.ac.lk)

methodology and the experimental setup. Section IV presents the results of our study, section V contains an analysis of the obtained results as a discussion and section VI concludes the paper.

II. RELATED WORK

In recent years many researches were conducted on forecasting with ANN. A brief analysis on a few of them are included here.

Kuligowski and Barros [4] used an ANN model and a linear regression model to forecast six hourly precipitation amounts on four locations in middle Atlantic region of the United States and found that the ANN model gives better results for heavy rainfall. They have used a dataset with 528 possible predictor variables and forward screening regression was used to select the best predictor variables.

Santhanam and Subhajini [5] evaluated the performance of RBFN with BPN to identify which ANN is the most effective on classification of rainfall prediction for Kanya Kumari district in India using 10 years meteorological data. They have classified the rainfall into rain and no rain. According to their study the RBFN was the most effective method with an accuracy of 88.5%.

Santhanam and Subhajini [6] has extended the same study by including GRNN in the performance analysis and evaluating the performance of GRNN, RBFN and BPN to identify which ANN is the most effective on classification of rainfall prediction. They have found that the GRNN was the most effective method with an accuracy of 96.8% and have outperformed BPN and RBFN in identifying both rain and no rain situations. The GRNN have a parallel structure which allow the network to train in a single iteration and as the size of the dataset increases the error approaches towards zero. These properties enable the GRNN to outperform RBFN.

Leung et al [7] used a GRNN to predict the monthly exchange rate of three currencies British Pound, Canadian Dollar and the Japanese Yen. Their results show that GRNN is an effective method for financial forecasting problems.

Luk et al [8] used a MLFFN to forecast short duration rainfall at specific locations within a catchment area. It had given reasonable prediction for the next 15 minutes but had difficulty in predicting the peak values of rainfall. They have found that the networks with simple structure yielded better performance and the networks with more hidden nodes tended to over learn the training data.

Lee and Liu [9] developed an automatic system for weather information gathering, filtering and prediction. They have used an agent based approach where a mobile agent was used to gather data and a Fuzzy-Neuro system to predict rainfall. The Neural Network was trained using Back Propagation algorithm. They have only considered the rain depth and were not interested in the exact amount of rainfall.

Gheyas and Smith [10] developed an ensemble with a collection of GRNN for time series prediction. They have used synthetic and real datasets to test their model. They found that the ensemble method gives better predictions compared with BPN and GRNN and other statistical forecasting methods.

Maqsood, Khan and Abraham [11] developed an ensemble neural network model for weather prediction in southern Saskatchewan, Canada. In this model weights for each

network in the ensemble were determined dynamically from the respective certainties of the network outputs. 24 hour ahead forecasts were made for temperature, wind speed and relative humidity. The performance of the ensemble was compared with different types of single networks and statistical models and it was found that the ensemble model gives the best performance.

The above studies indicate that various types of ANN models yield reasonable results in forecasting applications and they outperform statistical models.

Even though there are many researches done on forecasting using ensemble techniques in weather forecasting no studies were found on predicting the actual rainfall amount using ensembles. In this study the applicability of ensembles in forecasting the rainfall amount in Sri Lanka is investigated by developing different ENN models.

III. METHODOLOGY

A. Study Area

The study area was selected to be Colombo which is located on western coast of Sri Lanka on North latitude $6^{\circ} 55'$ and East longitude $79^{\circ} 52'$.

Rainfall of Sri Lanka is influenced by the monsoon winds from the Indian ocean. Rainfall is categorised into four climate seasons. First inter monsoon season from March to April, Southwest monsoon season from May to September, second inter monsoon season from October to November and Northeast monsoon from December to February. Rainfall occurs in three types- monsoonal, convectional and depressional. Monsoon rain occurs during the two monsoon periods and is responsible for nearly 55% of the annual precipitation. The other types of rainfall occurs in inter-monsoon periods.

Colombo has a tropical monsoon climate. Although Colombo experience rain in all four seasons heavy rains occur from May to August from the Southwest monsoon and October to February. The annual rainfall of Colombo is about 240 cm.

B. Data Collection

The data used for predictor variables was NCEP_1961-2001 dataset. The dataset contains 41 years daily observed data from 1961 to 2001, derived from the NCEP reanalysis [12]. It contains 26 variables as described in Table 1.

The data set was obtained from the Canadian Climate Change Scenarios website (<http://www.cccsn.ca>) from the grid box 22 X, 32 Y where the middle of the grid box corresponds to North latitude $7^{\circ} 5'$ and East longitude $78^{\circ} 75'$. The daily NCEP values are the average of 4 values taken at 0Z, 6Z, 12Z and 18Z. (Z time is in reference to 0° longitude at Greenwich, England.)

Daily rainfall data for Colombo for 41 years (1961-2001) have been collected from the Department of Meteorology Sri Lanka (<http://www.meteo.gov.lk/>). This dataset was used as the output of the ANN.

The ANN models were trained using the first 25 years (1961-1985) data as training data. They were validated using the next eight years (1986-1993) data as validation data and

were tested using the remaining eight years (1994-2001) data as test data.

TABLE I
PREDICTOR VARIABLES

Variable	Description	Variable	Description
Mslpas	Mean sea level pressure	p5zhas	500 hpa divergence
p_fas	Surface airflow strength	p8_fas	850 hpa airflow strength
p_uas	Surface zonal velocity	p8_uas	850 hpa zonal velocity
p_vas	Surface meridional velocity	p8_vas	850 hpa meridional velocity
p_zas	Surface velocity	p8_zas	850 hpa velocity
p_thas	Surface wind direction	p850as	850 hpa geopotential height
p_zhas	Surface divergence	p8thas	850 hpa wind direction
p5_fas	500 hpa airflow strength	p8zhas	850 hpa divergence
p5_uas	500 hpa zonal velocity	r500as	Relative humidity at 500 hpa
p5_vas	500 hpa meridional velocity	r850as	Relative humidity at 850 hpa
p5_zas	500 hpa velocity	rhumas	Near surface relative humidity
p500as	500 hpa geopotential height	shumas	Surface specific humidity
p5thas	500 hpa wind direction	tempas	Mean temperature at 2m

C. Predictor Variable Selection

The NCEP_1961-2001 dataset was normalised over the complete period. i.e. the mean and standard deviation for the period were calculated and the mean subtracted from each daily value before dividing by the standard deviation. The predictor variables were in different ranges. The ANN can learn faster and find weights in a predictable range if all inputs are in similar ranges. To equalise the importance of predictor variables all predictor variables were scaled to a range from -1 to 1.

To chose the predictor variables first correlation analysis was performed. The Pearson correlation coefficient of Rainfall and each variable in the dataset were calculated. As summarised in Table II the correlation coefficients were very small.

Then principle component analysis was performed. According to Eigen analysis it was decided to use 8 components to represent 80.1%. The correlations between the variables and the components were small with a very few larger values making it difficult to identify the principle components that represent the dataset well. Finally it was

decided to use all 26 variables as predictor variables. The results of principle component analysis is given in Appendix A. Table VII shows the Eigen analysis of the Correlation Matrix and Table VII shows the correlations between the variable and the components.

TABLE II
CORRELATION COEFFICIENTS OF RAINFALL AND THE OTHER VARIABLES

Variable	Correlation Coefficient	Variable	Correlation Coefficient
Mslpas	-0.078	p5zhas	-0.063
p_fas	-0.094	p8_fas	-0.041
p_uas	-0.002	p8_uas	-0.011
p_vas	0.061	p8_vas	0.040
p_zas	0.068	p8_zas	0.074
p_thas	-0.028	p850as	-0.072
p_zhas	-0.056	p8thas	-0.027
p5_fas	0.048	p8zhas	-0.044
p5_uas	-0.021	r500as	0.105
p5_vas	0.060	r850as	0.113
p5_zas	0.076	rhumas	0.208
p500as	-0.040	shumas	0.158
p5thas	0.032	tempas	-0.039

D. Artificial Neural Network Methodology

An ANN is a computational model motivated by the biological neural networks. They consist of a large number of processing elements that can work in parallel. ANNs are mostly used in classification (pattern recognition) and prediction problems because they can derive meaning from data that are too complex to be handled by humans. ANNs can manipulate large volumes of data with noise and imprecise information easily. Due to these characteristics of ANN they will be ideal to handle the complex and imprecise weather data and provide an accurate rainfall prediction.

ANNs have to be trained, validated and tested before using in an application. Training is the process of adjusting network parameters to represent the data set. Once trained the network can give the output for the given set of input data it has not seen previously. This is called generalisation. When training the network there is a chance of the network adjusting its parameters to only match the training data set. This is called over fitting. Validation data is used to avoid over fitting and stop the training at an appropriate time. Testing is the process of checking the network to see whether it can match an unseen data set.

E. Multi Layer Feed Forward Network with Back

Propagation Algorithm

MLFFN is a basic ANN architecture with at least one hidden layer. The hidden layers provide nonlinearities to the network. The basic principal of an MLFFN is that all the connections point in one direction so that the data flow from input layer to the output layer. BPN is a kind of gradient descent technique with backward error propagation. The initial

network output is compared with the expected output and the network parameters are adjusted until the error becomes minimal. The performance of the BPN increases with the size of the data set available. One major limitation in BPN is that it is prone converge to local minima.

MLFFN with varying architectures were implemented by changing the number of hidden layers (one, two), number of nodes per layer (5, 6, 7, 8, 9, 10, 11, 12 in the first hidden layer and 3, 4 in the second layer) and the activation functions for the hidden and the output layers (Sigmoid (1) and Gaussian (2)). The input layer had 26 nodes one for each predictor variable and the output layer had one node. To train the networks BPN algorithm was used. Fig 1, depicts the architecture of the BPN.

$$f(x) = \frac{1}{1+e^{-4x}} \quad (1)$$

$$f(x) = \exp[-x^2/4] \quad (2)$$

The weights on the hidden and the output layer were calculated according to the following.

$$W_{ij}(t+1) = W_{ij}(t) + \eta \delta_{pj} x_i \quad (3)$$

Where the learning rate $\eta = 0.7$

The MLFN was trained until the Root Mean Square Error (RMSE) of training set was less than 0.1.

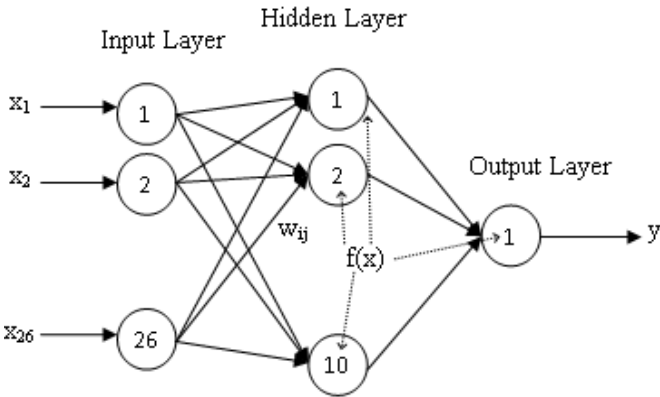


Fig 1. Architecture of the Multi Layer Feed Forward Network with Back Propagation Algorithm.

F. Radial Basis Function Network

A RBFN is a three layer feed forward network whose output units form a linear combination of the basis functions computed by the hidden units. The basis function (activation function) in the hidden layer produces a localised response to the inputs. The most common basis function used is the Gaussian function [13]. The network complexity and its generalisation capability depend on the number of neurons in the hidden layer. Learning in the RBFN can be divided into two stages, learning in the hidden layer using an unsupervised learning method followed by learning in the output layer using a supervised learning method [13]. The RBFN can be used for both classification and function approximation.

RBFN with three layers 26 nodes in the input layer one for each predictor variable and one node in the output layer was developed. The number of nodes in the hidden layer was varied methodically to find the number of nodes that gave the best prediction. The hidden layer was trained using k-means clustering algorithm and the output layer was trained using gradient descent algorithm.

In k-means clustering algorithm the centre of each cluster was initialized to a different randomly selected training pattern. Then each training pattern was assigned to the nearest cluster by calculating the Euclidean distances between the training patterns and the cluster centres. When all training patterns were assigned, the average position for each cluster centre was calculated. They then become new cluster centres. This process was repeated until the cluster centres do not change during the subsequent iterations.

Fig 2, depicts the architecture of the RBFN. The Gaussian function (4) was used as the RBF in the hidden layer.

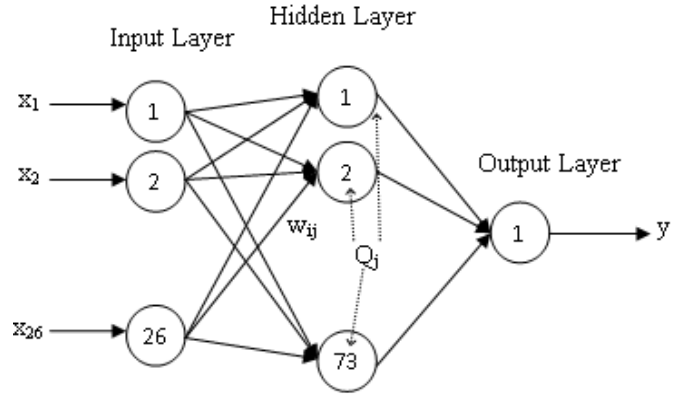


Fig 2. Architecture of the Radial Basis Network.

$$Q_j = \exp[-(X-W_j)^*(X-W_j)/2\sigma_j^2] \quad (4)$$

Where σ is the normalisation factor

Equation (3) was used to calculate the weights on the output layer with the learning rate $\eta = 0.7$

G. General Regression Neural Network

GRNN is a probabilistic neural network proposed by D. F. Specht [14]. GRNN is popular in forecasting applications because there is no iterative learning due to its parallel structure. The main advantage of GRNN is that it can learn in one iteration. Another advantage is that they can converge to the underlying function with a very few training samples compared with BPA. As the data set grows the error becomes zero. Because of these characteristics GRNN are widely used in forecasting applications.

There are four layers in GRNN: input layer, hidden layer, summation layer and output layer. The input layer consists of one node per predictor variable. The hidden layer consists of one node per training sample. This layer computes the Euclidean distance of the test case from the nodes centre point. The summation layer has only two nodes: the denominator which adds up the weight values from the hidden layer and the numerator which adds up the weight values multiplied by the

actual target value for hidden neurons. The output layer consists of one node which divides the value of numerator node by the value of denominator node.

GRNN with four layers 26 nodes in the input layer one for each predictor variable, a node for each input pattern in the hidden layer, two nodes in the summation layer and one node in the output layer (26,9131,2,1) was developed (Fig 3).

The input layer receives the input vector and distributes the data to the pattern layer. The pattern layer calculates O_j the Euclidean distance of each case from the nodes centre point using (4). The numerator uses (5) to add the weight values multiplied by the actual target value for hidden neurons, the denominator uses (6) to add weight values from the hidden layer and the output node uses (7) to divide the value of numerator node by the value of denominator node.

$$S_j = \sum_i w_{ij} \theta_j \quad (5)$$

$$S_d = \sum_i \theta_j \quad (6)$$

$$Y_j = S_j/S_d \quad (7)$$

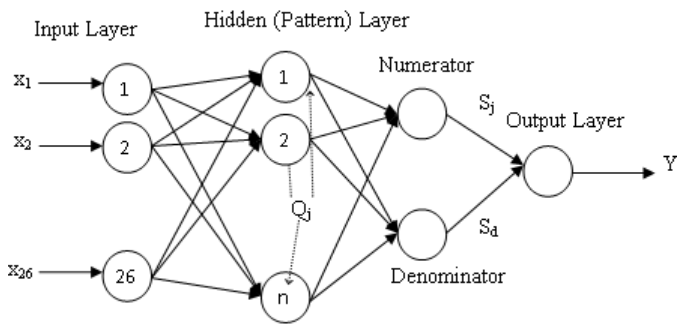


Fig 3. Architecture of the General Regression Neural Network

H. Neural Network Ensembles

Training a finite number of ANN for the same task and combining their results is known as an ANN ensemble. Hansen and Salamon's work [3] shows that the generalisation ability of ANN increase through ensemble. Due to this ANN ensemble techniques have become very popular in ANN applications. Creating an ensemble involves two phases: training the individual ANN and combining them.

1) Training the Individual Networks

There are many different ensemble techniques as explained by Sharkey [15]. These methods are based on varying the parameters related to the design and the training of ANN such as

- Varying the initial random weights - The ensemble is created by networks trained with different initial random weights.
- Varying the network architecture – The ensemble is created by networks with different network architecture. (Number of hidden layers, number of nodes in hidden layers, activation function)
- Varying the network type – The ensemble is created with different network types.

- Varying the training data – The ensemble is created with networks trained by different training data sets. (Different data sets obtained by sampling the training data, data sets obtained from different sources, data sets obtained by different preprocessing phases.)

Past researches [15], [16] have shown that combining these techniques can give better generalisation in the ensemble.

After training the set of ANN a decision has to be made about which ANN are to be included in the ensemble. There are various techniques available from trial and error methods to Genetic optimisation techniques like ADDEMUP [17] for this.

In this study two ensembles were created by combining some of the above techniques.

- ENN1 – Varying the network architecture and varying the initial weights.
- ENN2 – Varying the network type and varying the initial weights.

2) Combining the Networks

To combine the selected ANN the most frequently used methods are average or weighted average method. The simple average method gives the same priority for all ANN. It does not consider that some ANN are more accurate than others. In weighted average method each ANN is assigned a weight to minimise the Mean Squared Error (MSE) of the ensemble.

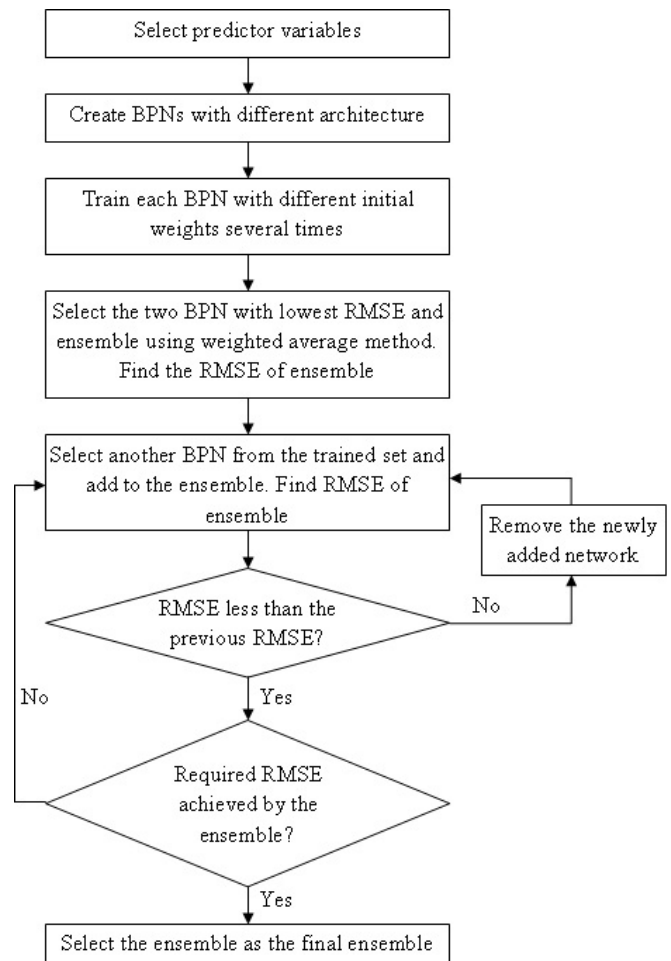


Fig 4. Flow chart for creating ENN1

ENN1 was created with 12 BPN created with different network architectures. The network architecture was changed by changing the number of hidden layers (one, two), number of nodes per layer (5, 6, 7, 8, 9, 10, 11, 12 in the first hidden layer and 3, 4 in the second layer) and the activation functions for the hidden and the output layers (Sigmoid (1) and Gaussian (2)). Each different BPN was trained several times with different initial weights. Fig. 4 illustrates the creation of ENN1 using a flow chart.

BPN, RBFN and GRNN models each shows different advantages and different generalisation capabilities. To incorporate all these advantages and to improve generalisation of the ensemble it was decided to use BPN, RBFN and GRNN models to create the ENN2. The networks were trained using different random initial weights and using the same training and validation datasets. Fig. 5 illustrates the creation of ENN2 using a flow chart. The architecture of ENN2 is illustrated in Fig. 6

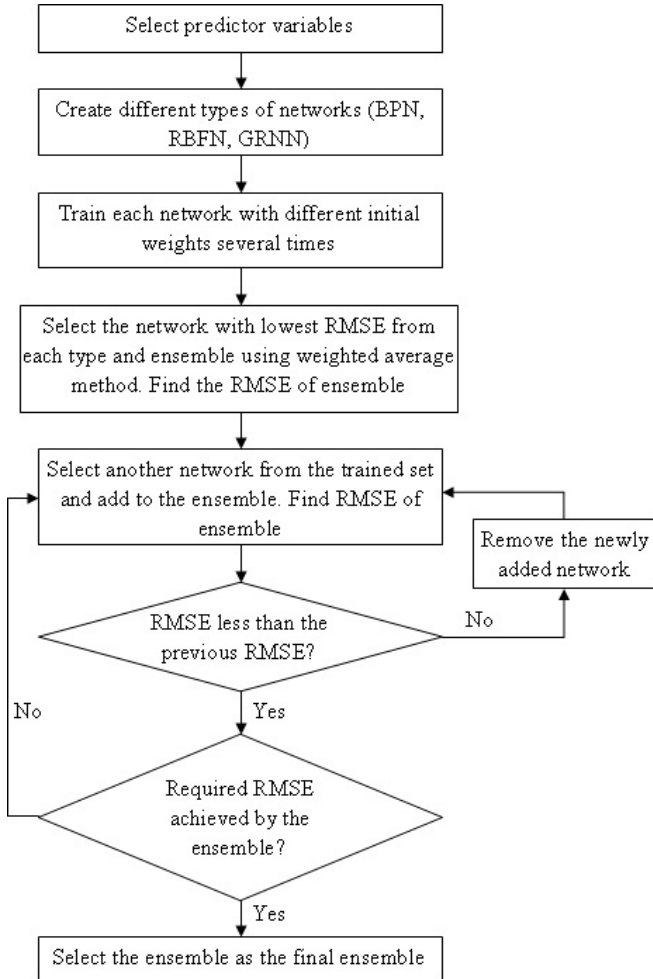


Fig 5. Flow chart for creating ENN2

For both ENN1 and ENN2 trial and error method was used to identify the ANNs that generate the best performing ensemble by trying out different combinations of the trained ANNs.

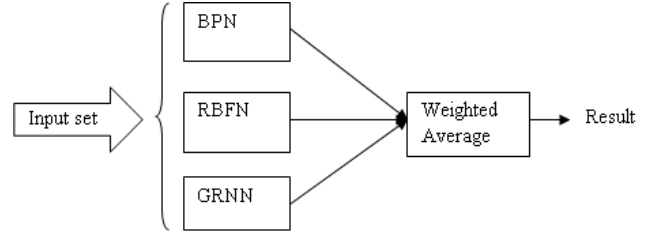


Fig 6. Architecture of ENN2

ANN were combined using the weighted average method. Weights were assigned to minimise the MSE of the ensemble. The validation dataset was provided for the ensemble and for each ANN and the MSE was calculated separately. The weights were assigned according to each MSE such that the summation of all weights is equal to one (8).

$$\frac{(\sum_1^n MSE_i)/MSE_i}{\sum_1^n ((\sum_1^n MSE_i)/MSE_i)} \quad (8)$$

Each previously trained BPN, RBFN and GRNN provided a separate prediction and those predictions were combined using the weighted average method to get the final forecast of the ensemble.

I. Measuring Forecasting Accuracy

Accuracy of a forecasting model is how well the forecasting model is able to reproduce data that is already known. In this study the Root Mean Square Error (RMSE) (9), Mean Absolute Error (MAE) (10) and the Coefficient of Determination R^2 are used as measurements of accuracy.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n e_i^2}{n}} \quad (9)$$

$$MAE = \frac{\sum_{i=1}^n |e_i|}{n} \quad (10)$$

Where e is the difference between the actual value and the predicted value.

The smaller the RMSE and MAE value the model is more stable but if there are a few larger errors RMSE value will magnify those errors and as a result will be larger. In such a case MAE is a better measurement.

IV. RESULTS

All ANN models in the study were trained with daily data of 25 years (1961-1985) and validated with data of next eight years (1986-1993) and the trained networks were tested using the remaining eight years (1994-2001).

A. Training Results of the BPN Models

BPN models were created by varying their architecture by increasing the number of the nodes in the hidden layers from (26,5,1) to (26,12,4,1) and changing the activation functions. Each network was trained several times with different initial weights and it was noted that the RMSE value

for testing decreases when the number of nodes are increased and then it starts to increase again.

The best performance was given by a BPN with (26,10,1) architecture and Sigmoid function as the activation function. Also it was noted that for many instances the BPNs with Sigmoid function as the activation function performed better than the BPNs with Gaussian function as the activation function. The average RMSE values obtained by the each BPN for testing is summarised in the Table III

TABLE III
CORRELATION COEFFICIENTS OF RAINFALL AND THE OTHER VARIABLES

Number of Nodes		Average RMSE for Testing Activation Function	
Hidden Layer 1	Hidden Layer 2	Sigmoid	Gaussian
5	-	9.87	9.85
	3	9.92	9.97
	4	9.95	9.93
6	-	9.84	9.78
	3	9.88	9.91
	4	9.91	9.86
7	-	9.76	9.77
	3	9.82	9.84
	4	9.80	9.88
8	-	9.70	9.73
	3	9.77	9.81
	4	9.73	9.84
9	-	9.64	9.68
	3	9.78	9.74
	4	9.69	9.71
10	-	9.51	9.62
	3	9.66	9.77
	4	9.82	9.86
11	-	9.60	9.57
	3	9.81	9.93
	4	9.79	9.82
12	-	9.72	9.74
	3	9.83	9.85
	4	9.87	9.91

B. Training Results of the RBFN Models

The generalization power of the RBFN depends on the number of hidden nodes. To find the number of nodes in the hidden layer of RBFN that gives the minimum RMSE, the number of hidden nodes in RBFN were incremented gradually and the networks were trained and tested several times using different initial weights.

Similar to the performance of BPN it was noted that after some variations the RMSE value for testing decreases when the number of nodes are increased and then it starts to increase again. The best performance was given by a RBFN with (26,73,1) architecture. The average RMSE values obtained by the each RBFN for testing is summarised in the Fig. 7.

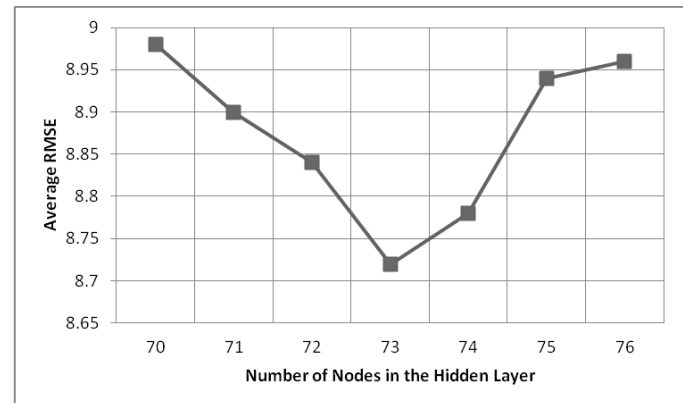


Fig 7. Average RMSE obtained by each RBFN for the Testing

C. Training Results of the GRNN Models

A GRNN with (26,9131,2,1) architecture was created and trained several times with different initial weights. It was noted that unlike BPN and RBFN the performance of the GRNN doesn't change much with the initial weights.

D. Training Results of the ENN1 Model

ENN1 was created by combining some of the trained BPN models using the weighted average method. The ensemble was started from two networks and networks were added one by one. The performance of the ENN1 when the networks are added is illustrated in Fig. 8.

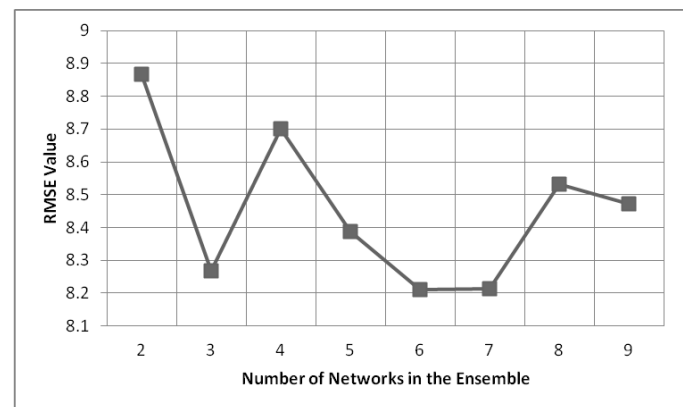


Fig 8. RMSE of ENN1 with the Number Networks in the ENN1.

The best performance was obtained by ENN1 when the number of networks were six and seven. The RMSE obtained in both cases was 8.21.

E. Training Results of the ENN2 Model

ENN2 was created by combining some of the trained BPN, RBFN and the GRNN models using the weighted average method. The ensemble was started from two networks and networks were added one by one to it using trial and error

method. The best performance was obtained by ENN2 when eight BPN, two RBFN and a GRNN model were in the ensemble. The RMSE obtained was 8.06. Table IV summarises some of the combinations of networks used in the ensemble.

TABLE IV
COMBINATIONS OF NETWORKS IN ENN2 AND THEIR RMSE

Number of Networks			RMSE
BPN	RBFN	GRNN	
8	2	1	8.06
7	2	1	8.07
6	2	1	8.09
8	1	1	8.12
7	3	1	8.14

Adding a BPN to the ensemble only made very small difference in the RMSE of the ensemble. Weights of each network in the best performing ENN2 is given in the Table V.

TABLE V
WEIGHTS OF THE NETWORKS IN ENN2

ANN Type	Individual RMSE After Training	Weight in the ENN2
BPN 1	9.44	0.03
BPN 2	9.48	0.03
BPN 3	9.49	0.02
BPN 4	9.52	0.02
BPN 5	9.57	0.01
BPN 6	9.60	0.01
BPN 7	9.61	0.01
BPN 8	9.63	0.01
RBFN 1	8.69	0.11
RBFN 2	8.86	0.14
GRNN	8.22	0.61

Table VI summaries the performance of each model for the test data by the RMSE value MAE value and the Coefficient of Determination (R^2).

TABLE VI
PERFORMANCE OF BPN, RBFN, GRNN, ENN1 AND ENN2

ANN Type	RMSE	MAE	R^2	Training Time (s)
BPN	9.44	5.36	0.47	14573
RBFN	8.69	5.04	0.55	573
GRNN	8.22	4.76	0.60	10
ENN1	8.21	4.74	0.60	-
ENN2	8.06	4.61	0.62	-

BPN gives the highest RMSE, MAE and the lowest R^2 and the ENN2 gives the lowest RMSE, MAE and the highest R^2 . The graphs of expected output vs. network output for BPN,

RBFN, GRNN, ENN1 and ENN2 are given in Appendix B Fig 8.

V. DISCUSSION

Compared with the performances of BPN, RBFN and GRNN the two ensemble models ENN1 and ENN2 showed better performance. From the two ensembles ENN2 showed the best performance. ENN1 was created with varying the network architecture and initial weights and ENN2 was created with varying the network type and initial weights. Results obtained by the study indicates that ensembles of networks gives better performance than individual networks and that varying the network type is more suitable for creating ensembles than varying the network architecture.

The ensemble models were able to predict almost all occurrences of zero rainfall accurately and were able to give reasonably accurate predictions for other rainfall occurrences. But for rainfall larger than 100 mm the predicted values by the ensembles were very small.

The RMSE value is larger than the MAE value by a considerable amount for all ANN models. This is because all models were unable to predict the higher rainfall accurately and these large errors were magnified by the RMSE value.

BPN was able to predict smaller rainfall values accurately but was unable to predict higher rainfall accurately. Higher rainfall predictions from GRNN were more accurate than that of RBFN. BPN, RBFN and GRNN were all able to predict the occurrences of zero rainfall accurately with RBFN giving the best performance but all performed poorly in predicting rainfall larger than 100 mm.

VI. CONCLUSION AND FUTURE WORK

Although the ensemble model predicts the rainfall with reasonable accuracy there are still deviations between the ensemble predictions and actual rainfall for higher rainfall occurrences. These high rainfall occurrences maybe the result of some random changes in the atmosphere.

In this study the ensembles were created by changing the network architecture and initial network weight (ENN1) and network type and initial network weights (ENN2). The ANN for the ensemble was selected using trial and error method and the weighted average method was used to combine the selected ANN. Further investigation on obtaining more accurate prediction for higher rainfall is to be done in future studies by experimenting with other ensemble techniques like changing the training data and network selection and combining methods.

ANN has become a widely used technique in weather prediction because of their ability to manipulate complex data and deal with noise efficiently. In this study the performance of two ensemble networks ENN1 and ENN2 in rainfall forecasting in Colombo, Sri Lanka was studied and its performance was compared with BPN, RBFN and GRNN models. Daily observed data for 41 years was used to train, validate and test the models.

The objective of this study was to investigate the performance of ensemble networks in rainfall forecasting. The results show that ensemble models predict rainfall more accurately than individual BPN, RBFN and GRNN models.

The major weakness of both ensemble models are that they were unable to predict higher rainfall accurately.

APPENDIX A

Result of the principle component analysis is given below. Table VII shows the Eigen analysis of the Correlation Matrix.

TABLE VII
EIGEN ANALYSIS OF THE CORRELATION MATRIX

Eigen Value	Proportion	Cumulative
9.1131	0.338	0.338
3.2012	0.119	0.456
2.8366	0.105	0.561
1.8493	0.068	0.630
1.4728	0.055	0.684
1.1878	0.044	0.728
1.0192	0.038	0.766
0.9472	0.035	0.801
0.8275	0.031	0.832
0.7781	0.029	0.860
0.6545	0.024	0.885
0.5606	0.021	0.905

Eigen Value	Proportion	Cumulative
0.5049	0.019	0.924
0.4663	0.017	0.941
0.3431	0.013	0.954
0.2529	0.009	0.964
0.2351	0.009	0.972
0.2156	0.008	0.980
0.1564	0.006	0.986
0.1141	0.004	0.990
0.0784	0.003	0.993
0.0508	0.002	0.995
0.0478	0.002	0.997
0.0318	0.001	0.998
0.0262	0.001	0.999
0.0230	0.001	1.000
0.0058	0.000	1.000

According to Eigen analysis 8 components were used to represent 80.1% in the principle component analysis. Table VII shows the correlations between the variables and the components.

TABLE VIII
CORRELATIONS BETWEEN THE VARIABLES AND THE COMPONENTS

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Mslpas	-0.214	-0.148	-0.225	0.249	-0.291	0.023	-0.223	0.151
p_fas	0.226	-0.115	-0.063	0.193	-0.008	-0.296	-0.147	-0.066
p_uas	0.281	0.076	0.224	0.096	-0.032	-0.018	0.004	-0.055
p_vas	-0.287	0.164	-0.064	0.003	0.078	-0.032	0.135	0.106
p_zas	-0.129	0.025	0.324	-0.321	0.150	0.093	-0.159	0.137
p_thas	-0.230	-0.131	-0.260	-0.040	0.077	-0.046	-0.039	0.087
p_zhas	0.269	-0.225	0.045	-0.042	-0.072	0.029	-0.119	-0.089
p5_fas	0.004	-0.062	-0.065	-0.143	0.176	-0.736	0.142	0.116
p5_uas	0.220	0.096	-0.033	0.300	0.107	0.172	0.142	0.350
p5_vas	-0.107	0.410	0.038	0.224	0.088	-0.170	-0.240	-0.121
p5_zas	0.108	0.109	-0.090	-0.240	0.171	0.054	-0.577	0.364
p500as	-0.151	-0.128	0.277	0.250	-0.438	-0.164	-0.126	0.105
p5thas	-0.185	-0.079	0.021	-0.326	-0.129	-0.208	-0.167	-0.415
p5zhas	0.104	-0.419	-0.039	-0.220	-0.074	0.172	0.248	0.120
p8_fas	0.236	-0.051	-0.133	0.181	0.069	-0.372	0.024	0.080
p8_uas	0.295	0.107	0.016	0.182	-0.011	-0.037	0.042	-0.013
p8_vas	-0.243	0.288	0.063	0.097	0.070	0.018	0.131	-0.006
p8_zas	0.193	0.172	0.059	-0.145	-0.036	-0.011	-0.421	0.095
p850as	-0.236	-0.177	0.003	0.237	-0.341	-0.012	-0.195	0.173

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
p8thas	-0.250	-0.144	-0.067	-0.136	0.075	-0.106	-0.008	0.061
p8zhas	0.240	-0.301	-0.073	-0.115	-0.054	-0.003	-0.125	-0.006
r500as	0.136	0.182	-0.209	-0.126	-0.195	0.057	0.091	-0.120
r850as	0.045	0.197	-0.289	-0.139	-0.216	-0.077	0.082	0.039
rhumas	0.122	0.277	-0.148	-0.211	-0.442	0.006	0.061	-0.090
shumas	0.093	0.157	0.403	-0.164	-0.346	-0.113	0.105	0.014
tempas	-0.038	-0.116	0.526	0.031	0.093	-0.093	0.062	0.087

APPENDIX B

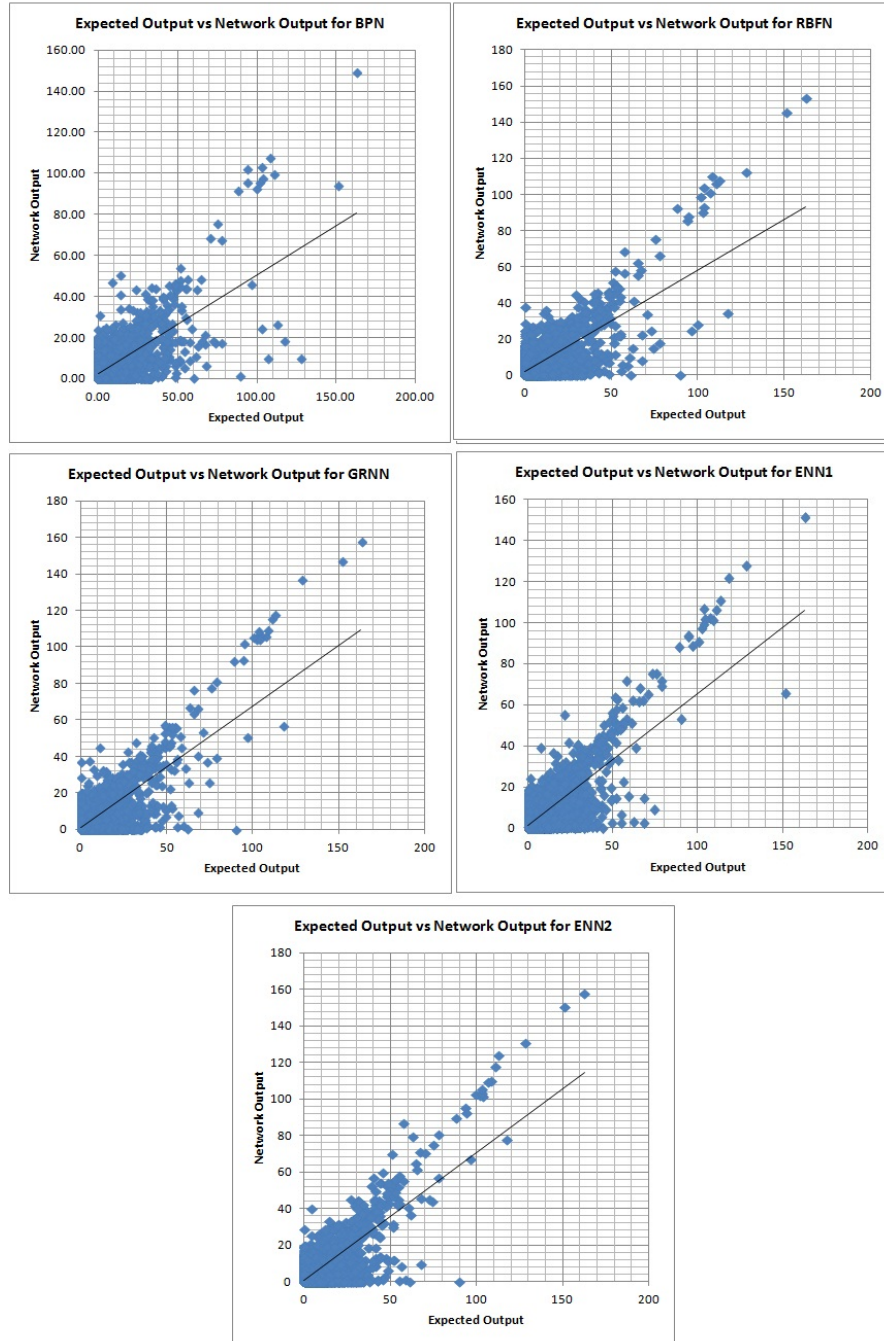


Fig 8. The graphs of expected output vs network output for BPN, RBFN, GRNN, ENN1 and ENN2 with the corresponding linear regression lines.

REFERENCES

- [1] Lynch P. (2008). The Origins of Computer Weather Prediction and Climatic Modelling. *Journal on Computational Physics*, 227: 3431–3444.
- [2] Malone T. F. (1955). Application of Statistical Methods in Weather Prediction. *Geophysics*, 41: 806–815.
- [3] Hansen L. K. and Salamon P. (1990). Neural network ensembles. *IEEE Trans Pattern Anal*, 12(10): 993-1001,
- [4] Kuligowski R. J. and Barros A. P. (1998). Localized Percipitation Forecasts from a Numerical Weather Prediction Model Using Artificial Neural Networks. *Journal on Weather and Forecasting*, 13: 1194-1204.
- [5] Santhanam T. and Subhajini A. C. (2011). An Efficient Weather Forecasting System Using Radial Basis Neural Network. *Journal of Computer Science*, 7: 962-966.
- [6] Santhanam T. and Subhajini A. C. (2011). Computational Performance of GRNN in Weather Freacsting. *Asian Journal on Information Technology*, 10: 165-169.
- [7] Leung M. T. Chen A. and Daouk H. (2000). Forecasting Exchange Rates Using General Regression Neural Networks. *Computers and Operations Research*, 27: 1093-1110.
- [8] Luk K. C. Ball J. E. and Sharma A. (2001). An Application of Artificial Neural Networks for Rainfall Forecasting. *Mathematical and Computer Modeling*, 33: 683-693.
- [9] Lee R. and Liu J. (2004). iJADE WeatherMAN: A Weather Forecasting System Using Intelligent Multiagent-Based Fuzzy Neuro Network, *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, 34(3): 369 – 377.
- [10] Gheyas I. A. and Smith L. S. (2009). A Neural Network Approach to Time Series Forecasting. *Proceedings of the World Congress on Engineering*, II.
- [11] Maqsood I. Khan M. R. and Abraham A. (2004). An Ensemble of Neural Networks for Weather Forecasting. *Neural Computing and Applications*, 13: 112 – 122.
- [12] Kalnay E., et al. (1996). The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77: 437-471.
- [13] Fu LiMin. (2003). *Neural Networks in Computer Intelligence*, Tata McGraw-Hill.
- [14] Specht D. F. (1991). A General Regression Neural Network. *IEEE Transactions on Neural Networks*, 2: 568-576.
- [15] Sharkey A. J. .C. (1999). *Combining Artificial Neural Nets - Ensemble and Modular Multi Net Systems*, Springer-Verlag.
- [16] Partridge D. (1996). Network generalization differences quantified. *Neural Networks*, 9(2): 263-271.
- [17] Optiz D. W. Shavlik J. W. (1996). Actively Searching for an Effective Neural Network Ensemble. *Connection Science*, 8: