Integration of Fuzzy C-Means and Artificial Neural Network for Short-Term Localized Rainfall Forecast in Tropical Climate

Noor Zuraidin Mohd-Safar[1], David Ndzi[1], David Sanders[1], Hassanuddin Mohamed Noor[1], Latifah Munirah Kamarudin[2]
[1]School of Engineering, University of Portsmouth, United Kingdom
[2]School of Computer and Communication Engineering, Universiti Malaysia Perlis, Malaysia
{noorzuraidin.mohdsafar, david.ndzi, david.sanders, hassanuddin.mohamednoor}@port.ac.uk
latifahmunirah@unimap.edu.my

***Abstract* - This paper evaluates the performance of a rainfall forecasting model. In this paper Artificial Neural Network (ANN) and Fuzzy C-Means (FCM) clustering algorithm are combined and used to forecast short-term localized rainfall in tropical climate. State forecast (raining or not raining) and value forecast (rain intensity) are tested using a number of trained networks. Different types of ANN structured were trained with a combination of multilayer perceptron with back propagation network. Levenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient training algorithm are used in the network training. Each neurons uses linear, logistic sigmoid and hyperbolic tangent sigmoid as transfer function. Input parameter preliminary analysis, data cleaning and FCM clustering were used to prepare input data for the ANN forecast model. Meteorological data such as atmospheric pressure, temperature, dew point, humidity and wind speed have been used as input parameters. The predicted rainfall forecast for 1 to 6 hour ahead are compared and analyzed. 1 hour ahead for state and value forecast yield high accuracy. Result shows that, the combined of FCM-ANN forecast model produces better accuracy compared to a basic ANN forecast model.**

Keywords-artificial neural network; fuzzy c-means; ANN; FCM; rainfall forecast; rainfall prediction; tropics; neural network; soft computing; meteorology; tropical climate; soft clustering

I. INTRODUCTION

Rainfall forecasting is extremely important for efficient water management and effective flood forecasting. Flooding is a common natural hazard in tropical countries that has significant impact on life and property [1]. Reliable rainfall forecasting will reduce the damages resulting from flood events, allowing time for evacuation and would form part of a flood management system. Rain in the tropics is characterized by high intensity over short periods of time with a smaller rain cell size that is localized and maybe highly dynamic depending on wind condition. Due to these characteristics, localized flash flood occurrence is frequent. In order to predict this catastrophic events, an accurate rainfall forecast is crucial. Rainfall forecasting remains one of the most challenging topic in hydrological operation due to spatial and temporal variations of rainfall distribution, dynamic behaviour of tropical climate and highly non-linear nature of rainfall events. Meteorological data and rainfall analysis from North-West Malaysia is selected to study and develop an appropriate model. Numerical Weather Prediction (NWP) method is commonly used for weather forecasting in Malaysia [2]. NWP forecasting was developed for large areas and it is therefore not suitable for localized rain and small rain cell sizes. NWP process requires exhaustive mathematical computation and a high performance computer with a large size storage capacity. Furthermore, NWP model used in forecasting is adapted from non-tropical regions such as Europe and Japan [2] and is therefore not relevant for local tropical weather forecasting because rainfall in temperate climate is widespread. Artificial Neural Network (ANN) is capable of learning the relationship between rain patterns and associated meteorological parameters in order to produce forecasting. Supervised learning techniques are applied during the training process to find adjusted weights that minimize the errors between measured and predicted value. The proposed ANN model in this paper uses atmospheric pressure, temperature, dew point, humidity and wind speed data measured at the same point as rainfall measurements. The permutation of meteorological parameters, expert knowledge and empirical study are used to process the data and develop the ANN model. Fuzzy C-Means (FCM) clustering algorithm is applied in order to find the relationship between the environmental parameters and rainfall rate. This paper will also determine the reliability of the clustering technique in forecasting rainfall rate by exploring the integration of FCM and the ANN model. The performance outcome from the proposed model is determined by evaluating the magnitude of the error and the measurement of the correlation between the observed and forecasted values. Results showed that 1 hour to 4 hours ahead forecasts yield good results. The results show that integrating FCM and ANN models provide an ability to forecast a rainfall event and its intensity. This paper is organized as follows: section II describes the literature review, section III describes the study area, followed by the methodology in section IV. Section V presents the results and discussions and, finally, section VI is the conclusions.

II. LITERATURE

A. *Related works*

ANN has been used in weather forecasting since the idea of imitating human neurons began. It is useful to summarize some practical application areas encountered in relation to ANN driven weather forecasting. [3] presents the use of ANN learning models to provide 24 hour forecasts for temperature, wind speed, and humidity for weather in southern Saskatchewan, Canada. The results from [3] show that statistical model is less accurate compared to ANN model coupled with radial basis function. Even though the results show that ANN can be used but the geo-location and climate are different in this study. Some studies for short term rain and weather forecasts using ANN have also been reported in [4], [5] and [6]. [4] implements ANN to forecast it is raining or not raining regardless of rainfall amount or its intensity. The proposed ANN model in [4] were compared to U.S Weather Bureau forecasted values. The ANN model was trained using 24 hour sea level pressure pattern to forecast rain for San Francisco bay for 12 hours and 24 hours ahead forecasting. The forecasted results show that an adaptive system using ANN pattern recognition has the ability to produce acceptable accuracy in raining and not raining forecasts without understand thoroughly the dynamics of the parameter pattern. A case study of tropical climate was setup in Bangkok, Thailand to improve the current rainfall forecast using ANN [5]. Forecast results for near real time forecasting for 1 to 3 hour ahead were achieve more than 0.7 correlation [5]. A combination of spatial meteorological parameters such as humidity, air pressure, temperature and cloud condition from 75 weather stations were used to train the network. Generalized feed-forward network and simple Multilayer Perceptron (MLP) network architecture were applied in the proposed model [5]. A similar study was conducted in the Philippines using daily meteorological parameters to train the network [6]. Experiments showed that the daily rainfall forecast achieved high accuracy when ANN and Bayesian network were implemented [6]. All of these forecasting methods are taking advantage of high spatial and temporal data. For example, data availability for the study in [5] came from 104 weather stations measured over a period of 15 years. The recent advances in ANN methodology for modelling non-linear and dynamical phenomena in a range of applications is the motivation to investigate the application of ANNs for the prediction of hourly rainfall state and intensity in a tropical climate.

B. Fuzzy C-Means clustering

FCM is an iterative and unsupervised algorithm initially developed in [7] and [8]. FCM uses fuzzy partitioning of dataset to position the data point that belong to multiple cluster groups with correspondence fuzzy truth value between 0 and 1 [9]. FCM is an iterative algorithm that will define the centres of clusters (centroids) from the data reciprocal distances in order to minimize dissimilarity. Assuming the matrix of dataset is U, FCM randomly initializes the membership matrix of the dataset based on (1).

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,...,n \tag{1}$$

The dissimilarity function which is used in FCM is given by (2):

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}{}^{m} d_{ij}{}^{2} \tag{2}$$

where u_{ij} is degree of membership of x_i which takes a value between 0 and 1 (for a given data point j, the sum of the membership values for all clusters is 1), c_i is the centroid of cluster i, with n is the number of clusters, d_{ij} is the Euclidian distance between i^{th} centroid(c_i) and j^{th} data point, and m is a weighting exponent (where $1 \le m \le \infty$) [10]. An iterative optimization algorithm is used to minimize the dissimilarity functions through calculating the centre vectors using (3) in which x_i is the i^{th} data point. This is followed by an update of the matrix (U^k to U^{k+1}) where k is the iteration step in (4).

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}{}^{m} x_i}{\sum_{j=1}^{n} u_{ij}{}^{m}} \tag{3}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{4}$$

The iteration will stop when $\{\| u_{ij}^{k+1} - u_{ij}^{k} \|\} \le \varepsilon$ where ε is a termination criterion that has been defined.

The FCM algorithm is composed of the following steps:

1. Randomly initialize the membership matrix (U=[u_{ij}]) by using (1). Initialize the number of clusters (c), weighting exponent (m), termination criterion (ε) and iteration limit.
2. Calculate the centroids (c_i) using (3).
3. Compute dissimilarity between centroids and data points using (2).
4. Update U^k to U^{k+1} using (4).
5. If $\{\| u_{ij}^{k+1} - u_{ij}^{k} \|\} \le \varepsilon$ then the iteration will terminate otherwise recalculating the centroids (c_i) using (3) in step 2.

FCM iteratively updates after refining the centroids and the membership value for each data point within a group of dataset. However, FCM does not ensure that it converges to an optimal solution because the cluster centres are randomly initialized in matrix U. Performance depends on the initial centroids selected and for a robust approach, FCM should execute several times with different initial centroids. Before input parameters are feed to an ANN forecast model, FCM clustering was implemented to obtain the relationship between input parameters and the desired output pattern. FCM was also used to carry out pre-classification task in data reduction.

C. Artificial Neural Network

The idea of ANN is to imitate the human neuron processes. ANN consists of interconnected group of artificial neurons connected together with weighted connections. Each neuron has the ability to store information and experience for future use. The feed-forward neural network introduced in early model of

ANN and the structure of the ANN model is known as perceptron. This model generally uses a single perceptron layer and input directly feed to the output. The drawback of this approach is that a perceptron cannot be trained to recognize many classes of patterns. Each neuron is composed of two parts; weighted coefficient and transfer function. It consists of a summing function with an internal threshold and weighted as depicted in Fig. 1.

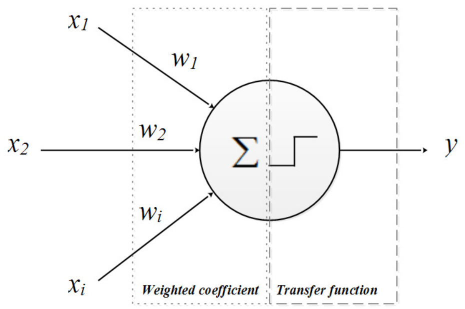


Fig. 1. Weighted input in single perceptron

For a neuron receiving n inputs, each input x_i (where i=1…n) is weighted by multiplying it by a weight (w_i) . The sum of the $w_i x_i$ products gives the net activation of the neuron. This activation value is subjected to a transfer function (f) to produce the neuron's output, y in (5).

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right) \tag{5}$$

Adding more perceptron layers to the network topology will increase the ability of the neural network to recognize more classes of patterns. The new approach is called Multi-Layer Perceptron (MLP). MLP works by adding more layers of nodes between input and output nodes. The neurons are arranged into an input layer, an output layer and one or more hidden layers. The learning process for MLP is known as Back Propagation Learning (BPL). The process repetitively calculates an error function for each input and back propagates the error from one layer to the previous layer. Neural network will learn from training data set that has input signals and its desired output known as target. Network training is an iterative process and the iteration coefficient of each node is modified using new data from a training data set. The transformation of weights depend on the following steps:

1. Each learning step starts with forcing input signal from a training data set.
2. Then the process will determine the output values for each neuron in each network layer.
3. Finally, based on the above step a weight is assigned to map the input with the desire output.

Fig. 2. illustrates the propagation of signal traversing through each neuron from input layer, hidden layer and finally the target. Supposed that the layer representation has L number of layers, including input, hidden and output layer, while l is representing the input layer and hidden layer that has N number of nodes in the form of $N(l)$, where $l=(0,1,\ldots, L; l=0$ *is the input parameter)* and $i=1,\ldots,N(l)$ is the node that has output from previous layer and it will be the input for the next layer. $y_{l,i}$ is the output that depends on incoming signals $x_{l,i}$ and parameters α, β, γ. Thus the following equation is the generalization of the output from each node:

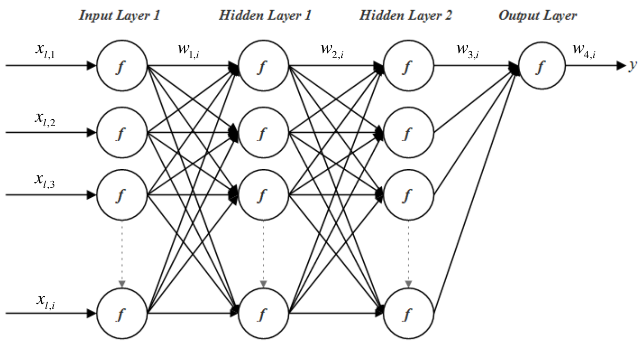


Fig. 2. ANN layer representation

$$y_{l,i} = f_{l,i}(x_{l-1,1}w_{l-1,1},\ldots,x_{l-1,N(l-1),}\alpha,\beta,\gamma,\ldots) \tag{6}$$

After the propagation of signals is done, the next step is to compare output signal from the node with the desired output (z) from training data. In general form the difference is the error signal $\delta_{l,i}$.

$$\delta_{l,i} = z - y_{l,i} \tag{7}$$

Assuming that the training data set has P entries, the error measured for the P^{th} entry of the training data set as the sum of the squared error:

$$E_P = \sum_{k=1}^{N(L)} (z - y_{l,i})^2 \tag{8}$$

where k is the number of components of z (desired output) and $y_{l,i}$ (predictive output). The weights for a particular node are adjusted in direct proportion to the error in the units to which it is connected by propagating it back through to all neurons by using Gradient descent algorithm. Gradient descent algorithm is responsible for finding the weights that will deliver the minimize error. The algorithm can be simplified into two steps as follows:

1. Obtain gradient vector.
2. Calculate the error signal $\varepsilon_{l,i}$ as the derivative of the error measure E_p with respect to the output node i in layer l in both direct and indirect paths. The ordered derivative can be express in the following equation:

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial y_{l,i}} \tag{9}$$

The whole process of training in MPL-BPL implementation is summarized in the following steps:

1. Initialize the weights using the training data by mapping the input data to the desired output data.
2. Initialize bias by randomly selected data from training dataset.
3. Compute the output of neurons, error and update the weights.
4. Update all weights and bias, then repeat step 3 for all training data
5. Repeat step 3 and step 4 until the error is reduce to an acceptable value.

III. STUDY AREA

Chuping is a small town in Perlis, Malaysia. It has 22,000 hectares of agricultural and plantation activities. The climate is tropical characterized by sunny days, high temperature and high humidity. In a whole year, the average day period is from 7 a.m. to 7 p.m. The average temperature is 27.5°C. Hourly meteorological data of atmospheric pressure, dry bulb temperature, dew point, humidity, wind speed, wind direction, rainfall amount and rainfall rate are measured and have been used to try to develop an ANN forecast model. Three years of data from 01/01/2012 to 31/12/2014 that consists of 26304 dataset have been used. From the available data, 2974 dataset are rainfall events and the remaining 23330 dataset are non-rainfall events. The ratio between non-zero rain data to zero rain data is about 11.3%. Elementary meteorological parameter characteristic for Chuping are summarized in Table I:

TABLE I. PARAMETER ANALYSIS FOR CHUPING

Parameter	Uni	Max	Mea	Min	Media	σ	σ^2
Pressure	*HPa*	1017.	1009.	1001.	1009.6	2.0	4.0
Temperatur	*°C*	38.1	27.5	16.1	26.4	3.2	10.2
Dew Point	*°C*	29.4	24.5	15.3	24.5	1.5	2.2
Humidity	*%*	100	83.7	29	88	12.	148.
Wind Speed	ms^{-1}	5.1	1.1	0	1.1	0.9	0.7

σ =Standard Deviation, σ^2=variance

IV. METHODOLOGY

A. Data cleaning

Data cleaning methods was used to eliminate noisy data. Three years of meteorological datasets are available from Chuping weather station. The rain gauge is a tipping bucket. Most of the cases, based on the distribution of rain, non-zero reading is stored in the weather station system. Some of the observed rain amount is less than 0.1 mm which have been considered as noise in this study. To eliminate noisy data, threshold and median filtering techniques are used to find the data outliers.

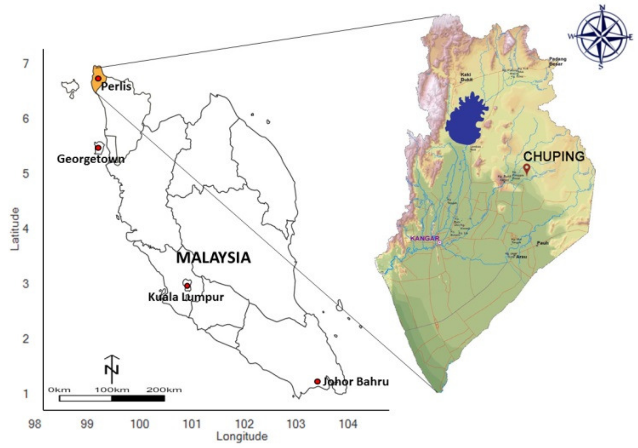


Fig. 3. Map of Malaysia and Chuping

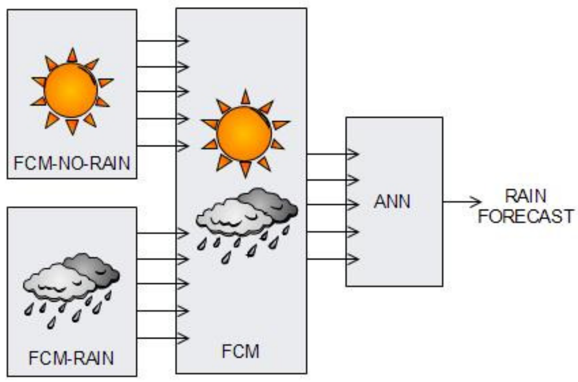


Fig. 4. Scematic illustration FCM-ANN rainfall forecast model

Then, FCM clustering method is used to find five clusters for each input. Fig. 4 depicted the schematic illustration of FCM-ANN implementation. Initially FCM is applied separately for zero rain and non-zero rain datasets. Then the clustered parameters are combined to create five input parameters and one target. The analysis to eliminate noisy data and cascade implementation of FCM for input parameters before they are fed to ANN forecast model determines the correct range for each input for zero rain and non-zero rainfall datasets. This process is important because if there is a similar class pattern between input parameters for rain and no rain conditions, FCM clustering implementation will reduce the ambiguity. Furthermore it can reduce overfitting during network training. Overfitting occurs when network training performs well on the training set but performs poorly on the test set.

B. ANN Design

Designing ANN model will follow the standard steps in ANN design from input data, train network and finally, use the network [11]. The following flow diagram is the step by step implementation in this study:

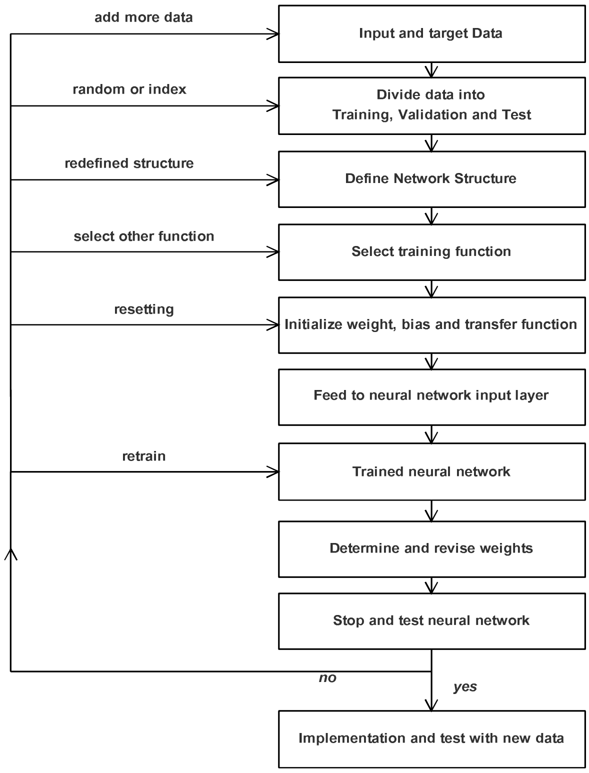


Fig. 5. ANN step by step design

1) ANN Preliminary implementation:

In the design stage, several models were tested in order to find the optimal ANN model. Collected data were divided into training, validating and testing sets:

- The training data was used to train the network. The error from each training step was used by the weight change algorithm to search finite weights that minimize the errors. Gradient descent algorithm was selected for this study.
- Validation data is used to periodically test the network as it trains to prevent overfitting.
- Test data is used when training has finished to test the generalization ability of the network for data it has never used in training or validation process.

In order to avoid overfitting and improve generalization, a small percentage of the training dataset was selected and used for cross validation. During the training process, error monitoring and validation is a necessity. When the error in the validation set increases, the training should be stopped because the point of best generalization has been reached. The cross validation approach with split-sample training was adopted for training the ANN models in this study. Three years of data with hourly observed parameters were selected to train the ANN models. Randomly selected data was divided into training, validation and test dataset with 70% for training, 15% for validation and 15% of the data for testing.

2) Data Normalization

It is important to pre-process input data before ANN training in order to minimize uncharacteristic inputs which may limit the application of the developed model [10]. A normalization method is used to pre-process input and target data. Data normalization is the process of scaling data to fall within a smaller range. One of the reasons for scaling is to equalize the importance of variables. The advantage of scaling data is to ensure that the weight of all neurons remain in a small, similar and within a predictable range. In this experiment, the normalization scale from -1.0 to 1.0 is adopted and the normalize data were derived from the following equation:

$$x'_{scale} = 2\left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) - 1 \tag{10}$$

where x is the value before normalization, $x_{\max}$ is the maximum value and $x_{\min}$ is the minimum value in the dataset.

3) Input layer, hidden layer and output layer

In the input layer, each neuron receives a single input that is a meteorological parameter. However, the hidden layer and output layer accept an arbitrary number of inputs based on the type of chosen interconnection of the neurons. Common practices to determine the number of hidden layers and neurons include trial and error method [12], genetic algorithm and Bayesian network modelling. Bayesian network modelling can be applied to determine the optimum neural network structure but this method is computationally complicated to implement [13]. There is also an input-output based guideline which suggests that the number of neurons in a hidden layer confirm to the following rule:

$$[2n^{1/2}, 2n+m] \tag{11}$$

where n is the number of inputs and m is the number of outputs in the neural network [14]. This study will use a combination of trial and error method and the range method from (11).

4) Training algorithm and transfer function

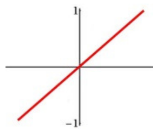
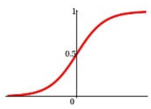
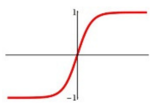
Every input parameter that is used to train the network is associated with the output pattern. Most of the ANN forecasting models use standard supervised learning MLP trained with BPL algorithm [15]. During the training process, comparative evaluation is made between the observed and forecast values to determine the smallest error. Even though the Gradient descent method is widely used in training algorithms, it is characterised by a slow convergence. Newer algorithms such as conjugate gradient technique have become more popular because they are faster [16]. Levenberg-Marquardt (LM), Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training algorithms are used selectively in this study. A summary of those algorithms are describe in Table II.

TABLE II. ANN TRAINING ALGORITHM.

Algorithm	Description
LM	Faster training algorithm for networks with moderate size, with memory reduction capability when the training data set is large. Using approximation method to update network weights and biases. LM algorithm for network training is used in [17] and [18].
BR	BR training algorithm improves generalization and reduces the difficulty of the determination of the optimum network architecture. BR uses object function such as MSE to improve generalization regularization technique requires expensive computation to reach optimal level.Detail discussion of BR is described in [19], [20] and [21].
SCG	SCG is designed based on optimization technique in numerical analysis named Conjugate Gradient Method [22]. Unlike other conjugate gradient approach SCG was designed to avoid time consuming line search, it may required more iteration to converge but the number of computation is reduced because line searched is avoided. This technique does not require any user specified parameters and its computation is faster and inexpensive. Detailed description of the algorithm can be found in [23].

Several combinations of the following transfer functions are used:

TABLE III. TRANSFER FUNCTION

Transfer function	Graphical illustration	Mathematical equation
Linear		$y = x$
Sigmoid (Logistic)		$y = \frac{1}{1+e^{-x}}$
Hyperbolic Tangent Sigmoid		$y = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

The transfer function is used to calculate the net weight or calculate a layer's output from its network input. Each layer in all experimental setup has its own transfer function. The optimum forecast results are based on combination settings of ANN learning algorithm and transfer functions. A summary of ANN model structure and its attributes will be discussed in the results and discussions section.

5) ANN architecture

The proposed ANN forecast model is based on MLP with BPL architectures. Different neural network structures have been tested. In general, the input parameters P_{input} and output parameters P_{output} for the neural network architecture can be described as follows:

$$P_{input} = [P_{t-n}, T_{t-n}, D_{t-n}, H_{t-n}, WS_{t-n}] \quad (12)$$

$$P_{output} = [R_t] \quad (13)$$

where R is the rainfall rate observed at time t and n is the number of hours before the rain. The ANN architecture can be simplified to:

$$[I - H_1 - ... H_n - O] \quad (14)$$

where I is the input layer, H is the hidden layer, O is the output layer and n is the number of hidden layers. For example, in the preliminary network architecture which consists of 5 inputs, 1 output and 2 hidden layers which have 20 and 10 neurons respectively, the architecture is 5-20-10-1.

C. Experimental setup

The experimental setup was divided into two main objectives; state forecast (raining or not raining) and value forecast (rain intensity). State forecast is a binary predictive target where 0 representing non-rainfall event and 1 is representing rainfall event while value forecast is the model that forecasts rainfall intensity. For the purposes of this study, the input vectors of atmospheric pressure, temperature, dew point, humidity and wind speed for the previous 2 to 7 hours were used for training and classification of ANN forecast model. Target vectors for state forecast is a binary number and target vectors for value forecast is rainfall rate. After data cleaning and clustering is implemented, hourly meteorological parameters from January 2012 to December 2014 and corresponding 1 to 6 hour target rain intensity were loaded into the data matrix. The number of hidden units in the single hidden layer was set to 10 for each model. The number of neurons and hidden layer are based on trial and error method and (11).

Each of the experimental setup finds the optimum outcome for hourly forecast, evaluates the error magnitude and correlation of the predicted output against the desired output. The initial numbers of experiments were performed using a basic ANN setup such as a single hidden layer, small amount of validation checks and training iterations (epochs). Randomly selected data for training, validation and test sometimes make the training to overfitting or saturation. ANN, by its nature, is biased towards the training dataset and will always produce high correlation value (high accuracy). In order to reduce the overfitting issue, the training dataset was small and indexing dataset selection approach was implemented.

D. Performance Indices

ANN model performance was evaluated using the following two evaluation parameters: Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*);

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad (15)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (16)$$

where y_t is the actual value, $\hat{y}_t$ is predicted value and n is the number of observations.

The correlation coefficient between observed and forecasted values of rainfall rate were measured. The correlation coefficient is a measure of the linear dependency and the strength of the relationship between two variables. If each variable has n scalar observations, then the Pearson correlation coefficient [24] is defined as:

$$R = \frac{n\sum y_t \hat{y}_t - \left(\sum y_t\right)\left(\sum \hat{y}_t\right)}{\sqrt{n\left(\sum y_t^{2}\right) - \left(\sum y_t\right)^{2}} - \sqrt{n\left(\sum \hat{y}_t^{2}\right) - \left(\sum \hat{y}_t\right)^{2}}} \quad (17)$$

The R value is an indication of the relationship between the forecasted and observed rainfall. If $R=1$, this indicates that there is an exact linear relationship and if R is close to zero, then there is non-linear relationship between them.

V. Results and Discussions

In general, the proposed ANN model has 5 inputs and 1 output for both state and value forecasting. Presented in this section are the best results from experimental setup for 1 to 6 hours forecasting based on calculated correlation using (17) and the error criteria using (15) and (16). Table IV shows the results of a combined FCM clustering method and the ANN model. Table V shows the results for a basic ANN implementation.

TABLE IV. Rain prediction using FCM-ANN

	Model	Transfer Function	Architecture	Training Function	MAE	RMSE	R
RAIN INTENSITY FORECAST	1HOUR	T,T,T	5-20-10-1	BR	0.4116	1.1166	0.8082
	2HOUR	T,T,L	5-10-5-1	BR	0.8122	1.6907	0.7052
	3HOUR	T,T,L	5-10-5-1	BR	0.9712	2.0038	0.6804
	4HOUR	T,T,T	5-10-10-1	BR	1.212	2.2384	0.5276
	5HOUR	T,T,T	5-10-10-1	BR	1.3973	2.4430	0.4910
	6HOUR	T,T,T	5-20-10-1	BR	1.5546	2.5916	0.4227
STATE FORECAST	1HOUR	T,T,T	5-10-1	LM	0.0767	0.1947	0.9204
	2HOUR	T,T,T	5-10-5-1	BR	0.1332	0.2571	0.8561
	3HOUR	T,T,T	5-10-5-1	BR	0.1764	0.2970	0.8023
	3HOUR	T,T,T	5-10-5-1	SCG	0.1830	0.3020	0.7940
	4HOUR	T,T,T	5-10-10-1	BR	0.2145	0.3265	0.7545
	5HOUR	T,T,T	5-20-10-1	BR	0.2543	0.3565	0.6975
	6HOUR	T,T,T	5-20-10-1	BR	0.2717	0.3686	0.6717

BR=Bayesian Regularization, LM= Levenberg Marquardt, SCG=Scaled Conjugate Gradient, T=Hyperbolic Tangent Sigmoid, L=Linear, S=Sigmoid

From the different learning algorithms used, BR proved to give a good convergence for forecasting. As shows in Table IV, for state prediction for 3 hour ahead the same configuration was used except the training algorithms were SCG and BR. The transfer function for ANN for each layer had a significant impact on the forecasting process. Hyperbolic Tangent Sigmoid and Linear transfer function fitted well in most of the ANN setup. Logistic Sigmoid did not meet the expected accuracy for prediction, presumed to be due to normalization of data. The number of hidden layers and its neurons are based on trial and error approach. When the number of hidden layers is set to more than 2 layers, the accuracy remains approximately the same but it slows down the convergence process. Therefore, the maximum number of hidden layers was set to 2. Estimating the rainfall intensity in all ANN models used between 5 to 20 neurons based on (11).

TABLE V. ANN Implementation

	Model	Transfer Function	Architecture	Training Function	MAE	RMSE	R
RAIN INTENSITY FORECAST	1HOUR	T,T,T	5-20-10-1	BR	0.3735	2.4079	0.3336
	2HOUR	T,T,L	5-10-5-1	BR	0.5420	2.4255	0.3054
	3HOUR	T,T,L	5-10-5-1	BR	0.6079	2.5361	0.0943
	4HOUR	T,T,T	5-10-10-1	BR	0.5983	2.5433	0.0572
STATE FORECAST	1HOUR	T,T,T	5-10-10-1	BR	0.0897	0.2124	0.5890
	2HOUR	T,T,L	5-10-5-1	BR	0.1174	0.2402	0.4072
	3HOUR	T,T,L	5-10-5-1	BR	0.1576	0.2848	0.3417
	4HOUR	T,T,L	5-10-10-1	BR	0.3095	0.2881	0.3095
	5HOUR	T,T,T	5-20-10-1	BR	0.1638	0.2912	0.2763
	6HOUR	T,T,T	5-20-10-1	BR	0.2250	0.3394	0.2687

BR=Bayesian Regularization, LM= Levenberg Marquardt, SCG=Scaled Conjugate Gradient, T=Hyperbolic Tangent Sigmoid, L=Linear, S=Sigmoid

State forecast provided better accuracy because ANN training easily adapted to the binary output. Fig. 6 shows the comparison between observed and forecast rainfall intensity and Fig. 7 is a regression plot for one hour ahead forecast. One hour ahead state forecast using FCM-ANN model gave 0.92 correlations and value forecast gave 0.80 correlation results. Rainfall rate forecast using the basic ANN model yielded much lower correlation value but only slightly worse MAE and RMSE for 1 to 4 hours forecast. The result shows that increasing the step ahead in hourly prediction will increases the error. Six hours ahead forecast can achieve 0.42 correlations for rainfall rate forecast and 0.67 correlations for state forecast using FCM-ANN. In basic ANN model implementation (Table V), rainfall rate forecast is less than 0.1 correlations value for more than 2 hours ahead. The correlation coefficient result is better in FCM-ANN model and it shows that the pattern of the rainfall forecast is following the expected output. For magnitude of error index, the values of RMSE are smaller for FCM-ANN model compared to basic ANN implementation. In contrast, MAE for ANN basic model yielded better results but the correlations is poor. The FCM-ANN model definitely produces a better solution for short-term forecast based on the correlations and RMSE results. The difference in accuracy results between FCM-ANN and basic ANN are tabulated in Table IV and Table V. As a result it can be concluded that FCM-ANN model produce better forecast for localized short-term rainfall in tropical climates.

VI. Conclusion

Rainfall is traditionally a very difficult weather phenomenon to predict, even using a large datasets spanning many decades [25]. In tropical area, rain dynamics are more challenging to predict because of their localized characteristic. The study area in this paper experiences high rain intensity over short durations and that are convective in nature. The highly dynamic and non-linear nature of this weather condition is a reflection of the complex interaction between the various processes at different scales. Therefore, the design of an effective system for prediction is difficult to realize. This study used ANN to develop a rainfall forecasting model. The existence data to train network for high intensity of rain patterns is fairly low which compounds the challenge to develop an accurate model. The best forecast model is 1 hour ahead forecast for state prediction using FCM-ANN. Using FCM clustering method to classified the ANN input parameters produces high accuracy in the ANN training. Comparative analysis between basic ANN implementation and FCM-ANN model clearly indicates that the performance level of FCM-ANN is better than that of a basic ANN model.

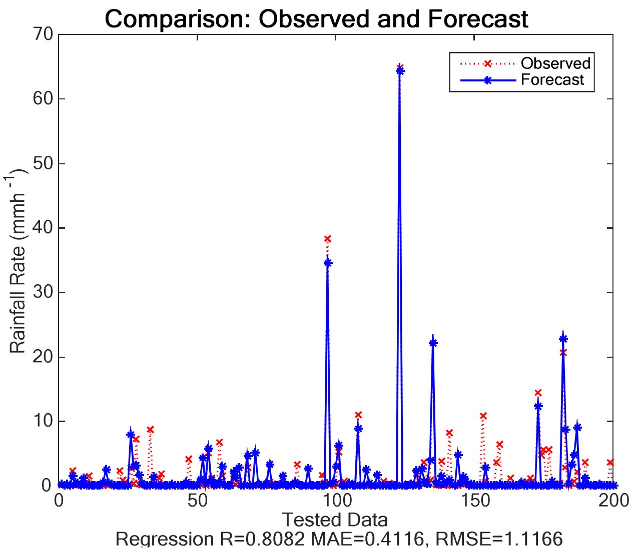


Fig. 6. Comparison between observed and forecast rainfall rate for 1 hour value forecast using FCM-ANN imlementation.

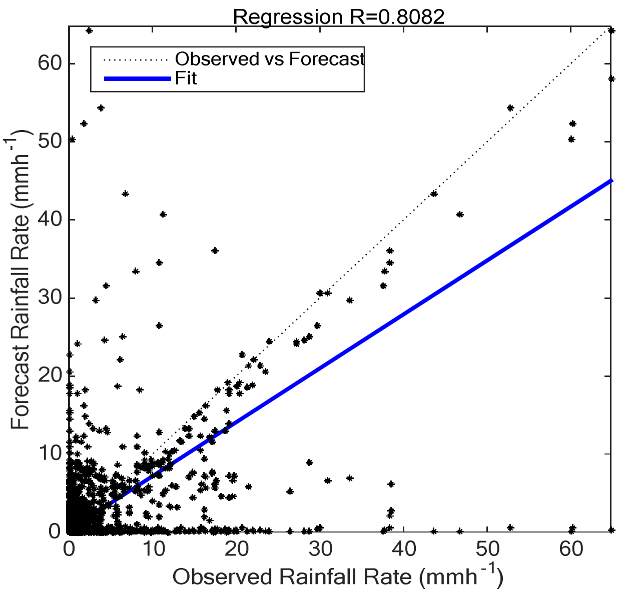


Fig. 7. Regression plot for 1 hour ahead value forecast for FCM-ANN imlementation.

REFERENCES

[1] B. Pradhan, 'Flood susceptible mapping and risk area delineation using logistic regression, GIS and remote sensing', *Journal of Spatial Hydrology*, vol. 9, pp. 1–18, 2009.

[2] A. Shahi, 'An Effective Fuzzy C-Mean and Type-2 Fuzzy', *Journal of Theoretical and Applied Information Technology*, pp. 556–567, 2009.

[3] I. Maqsood, M. Khan, and A. Abraham, 'An ensemble of neural networks for weather forecasting', *Neural Computing and Applications*, vol. 13, pp. 112–122, 2004.

[4] M. J. C. Hu and H. E. Root, 'An Adaptive Data Processing System for Weather Forecasting', *Journal of Applied Meteorology*, vol. 3, no. 5, pp. 513–523, Feb. 1964.

[5] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi, 'An artificial neural network model for rainfall forecasting in Bangkok, Thailand', *Hydrology and Earth System Sciences*, vol. 13, no. 8, pp. 1413–1425, 2008.

[6] C. N. A. Klent Gomez Abistado and E. A. Maravillas, 'Weather Forecasting Using Artificial Neural Network and Bayesian Network', *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 18, no. 5, pp. 812–817, 2014.

[7] J. C. Dunn, 'A Fuzzy Relative of the {ISODATA} Process and Its Use in Detecting Compact Well-Separated Clusters', *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, Jan. 1973.

[8] J. C. Bezdek, 'FCM:THE FUZZY c-MEANS CLUSTERING ALGORITHM', vol. 10, no. 2, pp. 191–203, 1984.

[9] Y. Lu, T. Ma, C. Yin, X. Xie, W. Tian, and S. Zhong, 'Implementation of the Fuzzy C-Means Clustering Algorithm in Meteorological Data', *International Journal of Database Theory and Application*, vol. 6, no. 6, pp. 1–18, 2013.

[10] A. Vega-corona, 'ANN and Fuzzy c-Means Applied to Environmental Pollution Prediction', pp. 1–6, 2012.

[11] M. B. Howard Demuth and M. Hagan, 'Neural network toolbox user's guide', 2012.

[12] 'Neural Networks Training Based on Differential Evolution Algorithm Compared with Other Architectures for Weather Forecasting34', *Journal of Computer Science and Network Security*, vol. vol 9 no 3, no. January, pp. 92–99, 2009.

[13] B. Cheng and D. M. Titterington, 'Neural Networks: A Review from a Statistical Perspective', *Statistical Sciences*, vol. 9, no. 1, pp. 2–30, 1994.

[14] D. F. E. Goss, 'Forecasting with neural networks: An application using bankruptcy data', *Information Management*, vol. 24, no. 3, pp. 159–167, Mar. 1993.

[15] A. Abraham, N. S. Philip, and K. B. Joseph, 'Will We Have a Wet Summer ? Soft Computing Models for Long-term Rainfall Forecasting', 1992.

[16] C. Charalambous, 'Conjugate gradient algorithm for efficient training of artificial neural networks', *IEE Proceedings G Circuits, Devices and Systems*, vol. 139, no. 3, p. 301, 1992.

[17] F. D. Foresee and M. T. Hagan, 'Gauss-Newton approximation to Bayesian learning', in *Proceedings of International Conference on Neural Networks ({ICNN}\textquotesingle}97)*.

[18] V. Pellakuri, D. Rajeswara Rao, P. Lakshmi Prasanna, and M. V. B. T. Santhi, 'A conceptual framework for approaching predictive modeling using multivariate regression analysis vs artificial neural network', *Journal of Theoretical and Applied Information Technology*, vol. 77, no. 2, pp. 287–290, 2015.

[19] J. J. Moré, 'The Levenberg-Marquardt algorithm: Implementation and theory', in *Lecture Notes in Mathematics*, Springer Science \mathplus Business Media, pp. 105–116.

[20] D. J. C. MacKay, 'A Practical Bayesian Framework for Backpropagation Networks', *Neural Computation*, vol. 4, no. 3, pp. 448–472, May 1992.

[21] D. J. C. MacKay, 'Bayesian Interpolation', *Neural Computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[22] J. R. Shewchuk, 'An Introduction to the Conjugate Gradient Method Without the Agonizing Pain', *Science*, vol. 49, no. CS-94–125, p. 64, 1994.

[23] M. F. Møller, 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Networks*, vol. 6, no. 4, pp. 525–533, Jan. 1993.

[24] J. Hauke and T. Kossowski, 'Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data', *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[25] H. L. Yuan, X. G. Gao, S. L. Mullen, S. Sorooshian, J. Du, and H. M. H. Juang, 'Calibration of probabilistic quantitative precipitation forecasts with an artificial neural network', *Weather and Forecasting*, vol. 22, no. 6, pp. 1287–1303, 2007.