IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 19, 2018, accepted October 26, 2018, date of publication November 9, 2018, date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2880044

Monthly Rainfall Forecasting Using One-Dimensional Deep Convolutional Neural Network

ALI HAIDAR AND BRIJESH VERMA
Central Queensland University, Sydney, NSW 2000, Australia
Corresponding author: Ali Haidar (a.haidar@cqu.edu.au)

This work was supported by the CQUniversity Research Division.

ABSTRACT Rainfall prediction targets the determination of rainfall conditions over a specific location. It is considered vital for the agricultural industry and other industries. In this paper, we propose a new forecasting method that uses a deep convolutional neural network (CNN) to predict monthly rainfall for a selected location in eastern Australia. To our knowledge, this is the first time applying a deep CNN in predicting monthly rainfall. The proposed approach was compared against the Australian Community Climate and Earth-System Simulator-Seasonal Prediction System (ACCESS), which is a forecasting model released by the Bureau of Meteorology. In addition, the CNN was compared against a conventional multi-layered perceptron (MLP). The better mean absolute error, root mean square error (RMSE), Pearson correlation (r), and Nash Suttcliff coefficient of efficiency values were obtained with the proposed CNN. A difference of 37.006 mm was obtained in terms of RMSE compared with ACCESS and 15.941 compared with conventional MLP. Further investigation revealed that the CNN was generally performing better in months with higher annual averages, while ACCESS was performing better in months with low annual averages. The generated output is promising and can be widely extended in this type of applications.

INDEX TERMS Convolutional neural networks, rainfall prediction, weather forecasting models.

I. INTRODUCTION

Weather forecasting ensures the sustainable development of society and economy. Therefore, the interest in forecasts has started since 650 BC, where Babylonians tried to predict weather based on observations of clouds (observed patterns). Then, multiple philosophers proposed various forecasting theories. Over time, it was noticed that these theories were not adequate. Consequently, it was perceived that there is a need to understand the weather from a broader perspective. With the invention of new instruments, measurement of the atmosphere was undertaken. Various instruments, such as the telegraph and radiosonde, allowed better monitoring of weather conditions. Nowadays, these instruments are used to record weather conditions. For modern rainfall forecasting, forecasts were produced before the invention of the computer, where Lewis Fry Richardson used arithmetic equations to predict weather after World War I (1922). Consequently, scientists introduced new methods that were developed along with the vast spread of technology. Nowadays, scientists use different methods to apply forecasts. Because to its relevance to human life and needs, weather forecasting is applied everywhere in the world.

Various weather forecasting mechanisms have been introduced and used in Australian rainfall prediction. Predictive Ocean Atmosphere Model for Australia (POAMA) was the recent official forecasting model used by Bureau of Meteorology (BOM) [1]. POAMA is based on General Circulation Models (GCMs) and is used to forecast various weather attributes, including temperature and precipitation [2], [3]. POAMA forecasts are given to users over large spatial distributions ($\approx$ 250 km grids) and as probabilistic values. Recently, the BOM revealed a new forecasting model for the Australian continent: the Australian Community Climate and Earth-System Simulator (ACCESS) [4]. This forecasting model consists of 11 ensemble members and releases forecasts up to six months. A lower grid size was used in this model ($\approx$ 60 km).

Artificial Neural Networks (ANNs) are machine learning algorithms representing a computational technology built on the analogy of the human information processing system [5]. These algorithms have been successful in various classification and regression tasks, such as financial applications [6], speech recognition, machine vision [7], [8], engineering applications [9], energy demand [10], medical

applications [11]–[13], and agricultural applications [14]. In addition, ANNs have been widely used in predicting weather attributes such as rainfall, temperature, relative humidity and wind [15]–[21]. Furthermore, it has been applied in anticipating several environmental phenomena as floods [22], drought indices [23], [24], water demand [25], heavy metal concentration in lakes [26], etc.

ANNs are computational paradigms that attempt to approximate non-linear relationships [27]. These models require data, where a set of input features is given to the network to approximate a non-linear relationship with the target. ANNs mimic the structure of the human biological brain. The most commonly used network topology in weather related applications is the Feed Forward Neural Network (FFNN), that processes input features in a feed forward manner. As mentioned earlier, a neural network requires data to be trained so that it can be used to estimate future weather conditions. The backpropagation algorithm is usually incorporated as the training algorithm, where the network connection weights and bias are updated to achieve the lowest error between actual and forecasted values.

The use of neural networks in weather applications have been applied for various lead times and in different locations around the world. This included hourly, daily, monthly, quarterly, annually etc. Luk, Ball, and Sharma proposed a neural network to forecast rainfall amount 15 minutes ahead for Paramatta catchment in Sydney, Australia [28]. Only rainfall values were used as input features in the prediction task. Multiple neural network topologies were introduced and evaluated where a Time Delay Neural Network (TDNN) revealed the highest accuracy.

Chaudhuri and Chattopadhyay [16] developed a FFNN to estimate maximum surface temperature and maximum relative humidity. Lagged values of each attribute were used to train two FFNNs. The Prediction Error (PE) statistical measurement was used to assess the performance of the developed models. The developed models were compared against single layered perceptron models, where lower errors were obtained with the proposed FFNNs.

Baawain, Nour, El-Din, and El-Din conducted an experiment that uses an ANN as a forecasting model to predict two El-Nino Southern Oscillation (ENSO) attributes: Nino 3.0 and Southern Oscillation Index (SOI) [29]. A Multi-Layered Perceptron (MLP) was designed for each attribute. The authors investigated the ability of MLP forecasting at various lead time (1-12 months). Pearson correlation (r) was used to assess the performance of each MLP. Results exposed that increasing the lead time decreased the prediction accuracy.

Chattopadhyay developed a FFNN to estimate average rainfall during summer-monsoon season in India [30]. The proposed network was compared to a Multiple Linear Regression (MLR) model where better accuracy was obtained with proposed FFNN.

Hung *et al.* [31] proposed a Generalized Feed Forward Neural Network (GFFNN) to forecast hourly rain in Bangkok, Thailand. The authors concluded that accuracies decreased while increasing the prediction lead time.

Nagahamulla *et al.* [32] used an ANN to predict seasonal monsoon precipitation in Srilanka. The developed models targeted estimating rainfall for four months: May, June, July and August where climate indices were used as possible predictors. Correlation analyses were used to determine the predictors of each month. The generated networks were compared against each other and the best accuracy was obtained in June.

Meknik *et al.* [33] utilized weather attributes to forecast spring rainfall in Victoria, Australia. A MLR and an ANN were utilized as the prediction models. Data were collected from nine weather stations. Better accuracy was recorded with the neural network in eight out of the nine stations compared to MLR.

Kashiwao *et al.* [34] developed a prediction model for hourly rainfall prediction in Japan. MLP and Radial Basis Function Neural Network (RBFNN) topologies were mainly investigated to build the prediction system. The developed neural network based models held reasonable accuracy but lower than the official forecasts accuracy release by Japan Meteorological Agency (JMA).

Vathsala and Koolagudi [20] utilized a MLP to predict peninsular Indian summer monsoon rainfall. Closed-itemset-generation-based association rule method was utilized for feature selection and K-means clustering for dimensionality reduction. The target of the MLP was to classify the type of rainfall to be encountered over the targeted location.

Doe and Şahin [23] used ANNs to forecast the Standardized Precipitation and Evapotranspiration Index (SPEI), which is a drought index. Global scale climate indices were used as predictors of SPEI. Some 30 ANNs were developed while varying the input features and network parameters. The accuracy of each ANN was calculated and one was selected as the best model. The authors confirmed the applicability of ANNs in estimating SPEI.

Karmakar *et al.* [35] developed a neural network model to estimate monsoon precipitation for a specified region in India. Wang and Sheng [36] proposed a Generalized Regression Neural Network (GRNN) to forecast yearly precipitation for Zhengzhou, China. Moustris *et al.* [37] conducted a study that uses artificial neural networks to predict rainfall for multiple months for selected locations in Greece. Charaniya and Dudul [38] proposed a Focused Time Delay Neural Network (FTDNN) to predict Indian monsoon rainfall. Indian Ocean Dipole (IOD) climate index and rainfall values were used as predictors. Khedhire [39] proposed a neural networks based approach to predict rainfall for a selected location in Canada.

In addition, several machine learning algorithms as Support Vector Machines (SVMs), Adaptive Network-Based Inference Systems (ANFISs), Extreme Learning Machine (ELM), Regression Trees (RT) and K-Nearest Neighbors (KNNs) have been used to predict various weather attributes [40]–[44]. Mekanik *et al.* [41] proposed an ANFIS model to estimate spring precipitation for various locations in Australia. Yaseen *et al.* [45] proposed a hybrid ANFIS model

to predict monthly rainfall for a catchment area in Malaysia. Sojitra *et al.* [46] developed daily rainfall forecasting models using ANFIS for Udaipur city in India. Jaedong and Jee-Hyong [40] proposed an SVM based prediction model to forecast hazardous weather conditions. Kusiak *et al.* [42] employed five different algorithms to forecast rain in a watershed basin at Oxford, Iowa. Yaseen *et al.* [47] designed an ELM to predict stream flow for a selected region in Iraq. Deo and Şahin [43] used ELMs to estimate a drought index in eastern Australia. Bagirov and Mahmood [44] compared several machine learning models including SVMS and RT in predicting monthly rainfall in Australia. Further applications that utilize machine learning algorithms in weather related applications can be found in [5] and [48]–[50].

Convolutional Neural Networks (CNNs) are network topologies basically used in machine vision applications. Using a CNN, the number of training weights in the network are reduced when compared to the fully connected networks, where portions of input features vector share the same set of weights. In 2012, Alexnet which is a convolutional neural network topology, achieved the lowest error in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [51]. Since then, CNNs were widely used and investigated in various applications including machine vision, health and financial applications [11], [12], [52]–[57].

In recent literature, various artificial neural network topologies were incorporated to predict weather attributes. In this work, we investigate the ability of deep convolutional neural networks in approximating monthly rainfall for a selected location in Australian areas. The main reason beyond incorporating a convolutional neural network is its proven ability in extracting complex relationships in input features. To our knowledge, this is considered the first time applying one-dimensional deep convolutional neural networks in predicting monthly rainfall. The paper presents the following research contributions:

1) A new method for building forecasting models based on convolutional neural networks is proposed
2) The ability of deep CNNs in predicting monthly rainfall for locally specified regions in Australia is investigated
3) The performance of CNN against the first version of ACCESS prediction model ACCESS-S1 is analyzed
4) The performance of CNN against a conventional MLP is analyzed
5) The prediction performance of the CNN over each month is analyzed

The rest of this paper is organized as follows: Section II describes the proposed forecasting approach. Section III lists the collected data. Section IV describes experimental setup and results. Comparative analysis are shown in Section V. A conclusion is drawn in Section VI.

II. PROPOSED APPROACH

In this study, we propose a deep convolutional neural network to forecast monthly rainfall for a selected location. One dimensional convolutional neural networks are explained first then the prediction model is shown.

A. ONE DIMENSIONAL CONVOLUTIONAL NEURAL NETWORKS

Typically, a CNN consists of various combinations of three main layers: convolutional layer, pooling layer, and fully connected layer. A general architecture of a deep convolutional neural network with two convolutional layers (Conv1, Conv2), two pooling layers (Pool1, Pool2) and a Fully Connected Layer (FCL) is shown in Figure. 1.

Algorithm 1 CNN Training Procedure

Input: Training dataset, Validation dataset
Output: Trained CNN
Initialize the network weights and bias
For each epoch:
    Process the records of the training data cases
    Compare the actual values to predicted values
    Calculate the loss function
    Backpropagate the error through the layers and adjust the network weights
    Check the validation dataset
        If better loss value obtained
            Save the network weights
        End
End
Return the trained CNN

The convolutional layer is the main building block of the convolutional neural network. Usually, the layers of the network are fully connected in which a neuron in the next layer is connected to all the neurons in the previous layer. While with convolutional layers, neurons are connected to local regions from the previous layer. We intend to use this type of connections in an attempt to enhance the determination of effective predictors of rainfall. Several characteristics affect the structure of the convolutional layer including: filter size, number of filters, stride, padding. The filter size is the local receptive field that represents a spatial area over the input features vector. As shown in Figure. 1, the convolutional layer is viewed as a stack of layers. The number of stacked layers is determined based on the number of filters specified. Stride determines the way the filter (local receptive field) moves through input. Padding is the process of adding/removing values to the dimension of the input so that the filter can smoothly traverse the whole dimension.

The convolutional neural network learns from data through its hierarchical structure of layers. Going deeper with layers, enhances the learning of complex relationships in input features. In rainfall prediction, the forecasting model performance is dependent on input features. For a set of input features $\delta = \{\delta^1, \delta^2, \ldots, \delta^n\}$, $1 \leq i \leq n$, n is the number of input features, the use of a subset $\beta(\beta \subset \delta)$ may reveal

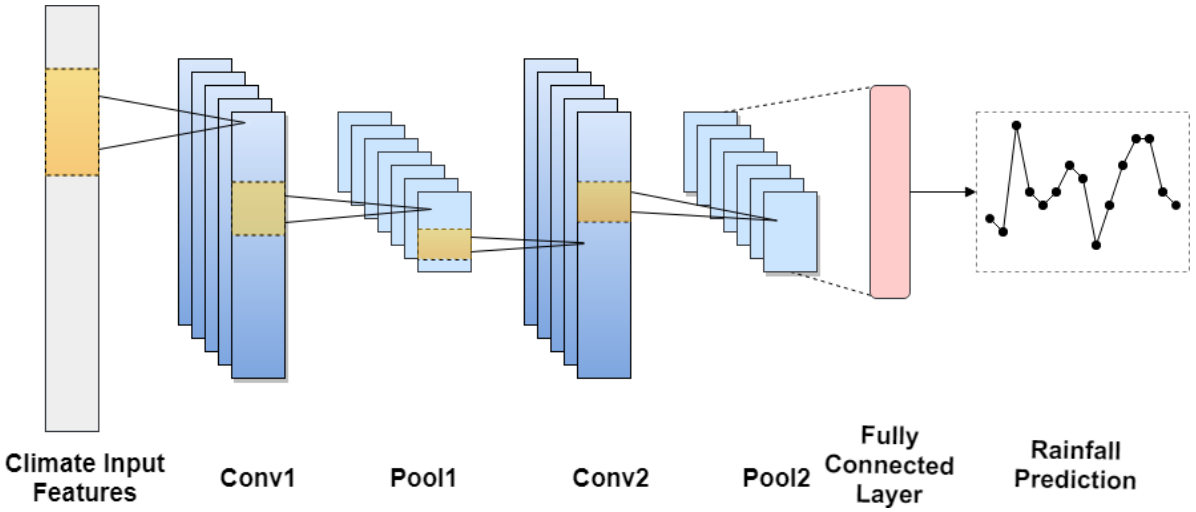

FIGURE 1. A general architecture of one-dimensional convolutional neural network.

better performance. On the other hand, the removal of elements in the input features dataset may remove some of the information needed for the model to perform. Also, a specific feature δ^i may highly affect the prediction of specific instances and may not be useful for others, especially in rainfall applications. Furthermore, the combination of multiple features through a set of mathematical representations may release better representations of the climate patterns. Hence, we propose using convolutional nets since the connections are directed toward specific regions in the input features vector. Through this mechanism, the deeper feature maps of the network could learn the relationships between input features and rainfall without the need to investigate all the features effectiveness on rainfall variability.

The pooling layer is added to decrease the size of the representations. The maximum pooling layer selects the highest value in each selected region of inputs while the average pooling layer selects the average value. Each convolutional layer was followed by a pooling layer when developing the network model.

CNNs have been widely used in classification applications. For this reason, different activation functions have been utilized to map the input features in a set of categories. In this study, the output is determined as rainfall amounts. For this reason, sigmoid (1) and hyperbolic tangent (2), which are a nonlinear activation functions, were investigated in the final layer:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

$$\sigma(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \tag{2}$$

where z is the summation of weights and inputs of the connected layer added to bias, $z = w_k x_k + b$, w_k represent a weight of a connection to the fully connected layer, x_k is an input into the fully connected layer, b is bias, $1 \leq k \leq c$, c is the number of connection weights and inputs in the fully connected layer. Additional layers are usually used as the flatten layer, which converts the input with multidimensional sizes into one dimension. Dropout is usually incorporated to avoid overfitting. The dataset is usually partitioned into training, validation and testing. The procedures followed to train the CNN are shown in Algorithm. 1. A set of functions can be deployed through the training phase. As shown in Algorithm. 1, a check point was introduced to save the network weights when having a better performance over the validation dataset.

B. MONTHLY RAINFALL PREDICTION

The proposed approach intends to investigate the CNN performance in a weather prediction task. In other meaning, the CNN is proposed to map a feature space $R^n \rightarrow R$:

$$\begin{Bmatrix} \delta_1^1 & \delta_1^2 & \dots & \delta_1^n \\ \delta_2^1 & \delta_2^2 & \dots & \delta_2^n \\ \vdots & \vdots & & \vdots \\ \delta_m^1 & \delta_m^2 & \dots & \delta_m^n \end{Bmatrix} \rightarrow \begin{Bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{Bmatrix} \tag{3}$$

where δ_i^j represents an instance of a climate input feature, r_i is a rainfall value, $1 \leq i \leq m$, $1 \leq j \leq n$, m is the length of the dataset, n is the number of input features.

Rainfall variability is affected by various atmospheric conditions. These conditions comprise of local and global variations in the atmosphere that lead to rain occurrences in a selected location. These conditions can be measured and are denoted as climate indices. Climate indices have been linked to weather variability in several locations in the world. These indices may affect rainfall variability at specific times of the year. In other words, a climate index can be used to analyse rainfall trends at certain durations of the year for a selected location. Therefore, several climate indices were collected and used as input features.

The target is to find the best connections and input weights that will reveal the lowest error between actual and predicted rainfall values. Each neural network is trained by updating the connection weights and bias in each layer. The training process continues until reaching a minimum error between the target and the model output values. Then, a forecasting model is obtained. While training, the network performance is measured. The loss function (l) that maps the set of input features into rainfall was determined as mean squared error:

$$l = \frac{1}{h}\sum_{i=1}^{h}\left(y_i - y_i^{'}\right)^2 \tag{4}$$

where y_i represents an actual instance in the dataset, $y_i^{'}$ represents a forecasted instance in the dataset, h is the length of the dataset. The forecasting model can be then represented as:

$$rain = \varphi\left(\delta\right) \tag{5}$$

where φ is the trained convolutional neural network, $\delta = \{\delta^1, \delta^2, \ldots, \delta^n\}$, δ^j is an input feature $1 \leq j \leq n$, n is the number of input features. Multiple input features were proposed in this study. The lagged values of some of these features were used as predictors of rain. Hence, the prediction of rain at time t can be represented as shown in (6).

$$rain_t = \varphi\left(\delta_{t-k}^1, \delta_{t-k}^2, \ldots., \delta_{t-k}^n\right) \tag{6}$$

where k is the lag value, $1 \leq k \leq 12$, 12 was selected as the last antecedent value for some of the features (the information about previous weather conditions in each record is up to one year), n is the number of input features, φ is the forecasting model (CNN). It should be noticed that k varied for each feature.

III. DATA

Innisfail is an Australian suburb located in north Queensland (17.522° S, 146.0285° E). The selected location is shown in Figure. 2. Innisfail receives an annual rainfall average that exceeds 3550 mm. Several weather attributes and climate indices were gathered from multiple sources to be used as rainfall predictors. Rainfall values, which were targeted in this study, were taken from the Bureau of Meteorology (BOM) [58]. Additional local attributes were also collected from the BOM as the mean minimum temperature (MinT) and the mean maximum temperature (MaxT).

Various climate indices that measure weather conditions in oceans around the Australian content were also collected and used to predict rainfall variability in Innisfail. These indices included: Southern Oscillation Index (SOI), Nino 1.2, Nino 3.0, Nino 3.4, Nino 4.0, Dipole Mode Index (DMI), Interdecadal Pacific Oscillation (IPO), Tripole Mode Index (TPI), North Pacific Index (NPI), North Atlantic Oscillation (NAO) and Pacific Decadal Oscillation (PDO). Furthermore, sunspot values were collected and used as predictors of rain. Additional details about climate indices can be found in [59]. The weather attributes were collected from five sources: BOM, Royal Netherlands Meteorological Institute Climate Explorer (KNMI) [60], Climate of the 20th century (C20C) [61], Earth System Research Laboratory (ESRL) [62] and Solar Influences Data Analysis Center (SIDC) [63]. The source, minimum, maximum, average, median, and standard deviation (STD) of each weather attribute are shown in TABLE 1.

FIGURE 2. Selected location.

TABLE 1. Source of each weather variable.

Attribute	Source	Minimum	Maximum	Average	Median	STD
Rainfall	BOM	0.000	2684.600	300.479	191.600	305.262
MaxT	BOM	21.800	33.900	27.936	28.200	2.597
MinT	BOM	10.400	25.800	19.341	19.700	3.069
SOI	BOM	-33.300	34.800	0.286	0.300	10.200
Nino1.2	KNMI	-2.530	4.379	-0.167	-0.290	0.927
Nino3.0	KNMI	-2.098	3.236	-0.109	-0.157	0.785
Nino3.4	KNMI	-2.159	2.512	-0.029	-0.057	0.755
Nino4.0	KNMI	-1.769	1.434	-0.028	-0.009	0.562
DMI	KNMI	-1.156	1.536	0.041	0.029	0.322
SUNSPOT	SIDC	0.000	359.400	90.663	74.200	72.887
IPO	C20C	-7.170	6.530	-0.045	-0.060	2.263
TPI	ESRL	-2.340	2.050	-0.119	-0.090	0.743
NAO	KNMI	-4.700	6.660	0.043	0.030	1.721
NPI	KNMI	996.440	1021.530	1012.646	1013.785	4.394
PDO	KNMI	-3.399	3.622	-0.158	-0.141	1.222

The collected attributes were manipulated so that the records range between January 1908 and December 2012. The collected rainfall records contained missing values. These values were replaced by values in a nearby weather stations or average. Neural networks perform better with small ranges. The minimum and maximum bounds varied between each weather attribute. Hence data were normalized to a smaller range between 0 and 1 following this equation:

$$d_i^{'} = \frac{d_i - max(d_{i=1}^m)}{max(d_{i=1}^m) - min(d_{i=1}^m)} \tag{7}$$

Where $d \in \delta$, d_i is an instance of the weather attribute, $max(d_{i=1}^m)$ is the upper bound of the weather attribute values, $min(d_{i=1}^m)$ is the lower bound of the weather attribute values,

$d_i^{'}$ is a normalized value, $1 \leq i \leq m$, m is the length of the weather attribute dataset.

To generate the dataset, lagged values of the collected attributes shown in TABLE 1 were used as possible predictors:

- Rainfall lagged values were added as input features. Some 12 input features representing the lagged values of rainfall up to one year ($rain_{t-1}$, $rain_{t-2}$, ..., $rain_{t-12}$) were generated to predict rain at time t
- Two new features were created to increase the identification of antecedent rainfall conditions. Those features were created to determine the type of rain (extreme/normal) in antecedent months. For a month at time t, two new binary features ($Blag_{t-1}$, $Blag_{t-12}$) were proposed to identify the occurrence of heavy rain at time t-1 and t-12
- For each weather attribute (MaxT, MinT, SOI, Nino1.2, Nino3.0, Nino3.4, Nino4.0, DMI, Sunspot, IPO, TPI, NAO, NPI, and PDO), two features were added to the dataset representing the attribute value at time t-1 and t-12

Therefore, the dataset consisted of 43 input features: 14 input features at time t-1, 14 input features at time t-12, 12 lagged rainfall values up to one year, 1 feature as month and 2 binary features to determine type of rain at time t-1 and t-12. TABLE 2 represents the inputs features of the generated dataset.

The generated dataset consisted of 104 years (Jan 1909-Dec 2012). The main reason behind starting in 1909 even if the values were collected from 1908 was to include lagged values as possible predictors. In other words, January 1909 is the first instance with completed lagged values.

The first 92 years (Jan 1909-Dec 2000) of the dataset were used for training and validating the convolutional neural network. 85 % were used for training and the remaining 15 % for validating the network performance and to avoid overfitting. The remaining 12 years (Jan 2001 and Dec 2012) were used to assess the model performance (hold-out sample).

IV. EXPERIMENTS AND RESULTS

A. EVALUATION METRICS

Determination of monthly rainfall values was targeted in this study. To assess the accuracy of the developed models, several statistical measurements that have been widely used in rainfall prediction tasks were calculated: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson correlation (r) and Nash Suttcliff coefficient of efficiency (NSE). The mathematical representation of each statistical measurement is shown in (8), (9), (10) and (11), as shown at the bottom of the next page, where y_i is the actual value, $y_i^{'}$ is the predicted value, $\bar{y}$ is the average of observed values, p is the number of elements in the dataset, $1 < i \leq p$. The MAE and RMSE values range between zero and infinity. The closer the values to zero the better the forecasts. Pearson correlation values range between negative one and positive one. The closer the values to one, the better the forecasts. The NSE values range between negative infinity and one. Negative NSE values indicate that the prediction model is not a better predictor than the mean of the actual values [64].

TABLE 2. Input features.

Nb	Attribute	Nb	Attribute	Nb	Attribute	Nb	Attribute
1	$Rain_{t-1}$	12	$Rain_{t-12}$	23	$Nino3.0_{t-1}$	34	IPO_{t-12}
2	$Rain_{t-2}$	13	$Blag_{t-1}$	24	$Nino3.0_{t-12}$	35	TPI_{t-1}
3	$Rain_{t-3}$	14	$Blag_{t-12}$	25	$Nino3.4_{t-1}$	36	TPI_{t-12}
4	$Rain_{t-4}$	15	$MaxT_{t-1}$	26	$Nino3.4_{t-12}$	37	NAO_{t-1}
5	$Rain_{t-5}$	16	$MaxT_{t-12}$	27	$Nino4.0_{t-1}$	38	NAO_{t-12}
6	$Rain_{t-6}$	17	$MinT_{t-1}$	28	$Nino4.0_{t-12}$	39	NPI_{t-1}
7	$Rain_{t-7}$	18	$MinT_{t-12}$	29	DMI_{t-1}	40	NPI_{t-12}
8	$Rain_{t-8}$	19	SOI_{t-1}	30	DMI_{t-12}	41	PDO_{t-1}
9	$Rain_{t-9}$	20	SOI_{t-12}	31	$Sunspot_{t-1}$	42	PDO_{t-12}
10	$Rain_{t-10}$	21	$Nino1.2_{t-1}$	32	$Sunspot_{t-12}$	43	Month
11	$Rain_{t-11}$	22	$Nino1.2_{t-12}$	33	IPO_{t-1}	-	-

B. EXPERIMENTAL SETUP AND RESULTS

The proposed approach was developed using Keras [65] package installed on top of Tensorflow framework [66]. Trial and error method was followed to determine the convolutional network architecture. Several architectures were developed and examined while varying the number of convolutional layers, number of pooling layers, filter sizes, number of filters, stride and activation functions. In addition, the optimizers of the loss function were varied to analyse the effect of each optimizer on performance. Three optimizers were examined: Adam, stochastic gradient descent and RMSprop. 1000 epochs were used to train each model.

The selected architecture consisted of two convolutional layers, two average pooling layers and a fully connected layer as shown in TABLE 3. Several steps were followed to avoid overfitting. Dropout, which is a regularization techniques was added to the architectures [67]. In addition, the network weights that were shown with the lowest error over the validation dataset were detained while training the CNN. A flatten layer was added before the fully connected layer to flatten the input into the fully connected layer so that one dimensional input vector is obtained. Hence, the convolutional layers were added in an attempt to extract information and understand patterns in local regions in the input features. Dropout was added before the fully connected layer. The loss function of the selected network architecture is shown in Figure. 3. The loss function over the training dataset was not extremely lower than the loss function over the validation dataset. This means that the network didn't overfit through the training phase.

TABLE 3. Selected convolutional neural network architecture.

Layer	Type	Parameters				
		Activation	Filter size	Nb Filters	Pool Size	Neurons
Conv1	Convolutional Layer	Relu	4	24	-	-
Pool1	Pooling Layer	-	-	-	2	-
Conv2	Convolutional Layer	Relu	2	32	-	-
Pool2	Pooling Layer	-	-	-	2	-
FCL	Fully Connected Layer (output layer)	Hyperbolic tangent	-	-	-	1

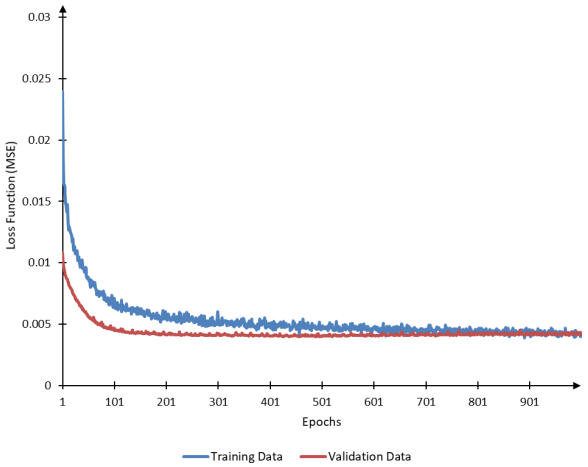

FIGURE 3. Loss function while training the convolutional neural network.

RMSEprop algorithm was found to be the best optimizer. Using a small number of filters didn't allow the network to converge. In addition, hyperbolic tangent activation function generated better models than sigmoid activation function. TABLE 4 shows the MAE, RMSE, and r values obtained by the developed CNN over the training, validation and testing datasets. The generated CNN model revealed a MAE of 114.654, RMSE of 142.133, and r of 0.868 over the test dataset. Because of dropout, it is expected to have the training error higher than the validation error since regularization techniques are turned off in validation and testing phases [65].

TABLE 4. Mae, RMSE, r values obtained over the training, validation and testing datasets.

Dataset	Training	Validation	Testing
Nb of records	938	166	144
MAE	124.823	124.947	114.654
RMSE	171.292	169.299	142.133
r	0.843	0.784	0.868

TABLE 5. Mae, RMSE, r and NSE for ACCESS, MLP and CNN forecasting models.

Model	MAE	RMSE	r	NSE
ACCESS-S1	125.345	179.139	0.763	0.575
MLP	120.674	158.074	0.843	0.669
CNN	114.654	142.133	0.868	0.732

V. COMPARATIVE ANALYSIS

The proposed approach was compared to ACCESS-S1 and MLP. A set of hindcasts from the ACCESS-S1 prediction model is available on the National Computational Infrastructure (NCI) website [68]. A dataset representing hindcasts between Jan-2001 and Dec-2012 was collected from the NCI website. These records are saved in Network Common Data Form (NetCDF) files and represent the output of ACCESS-S1 over Australia. As mentioned earlier, the set of forecasts are released over grid areas (60 km). MATLAB was used to collect the predictions released by ACCESS-S1 for the closest grid point to Innisfail with the following coordinates:

$$MAE = \frac{1}{p}\sum_{i=1}^{p}\left|y_i^{'} - y_i\right| \tag{8}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p}\left(y_i^{'} - y_i\right)^2}{p}} \tag{9}$$

$$r = \frac{p\left(\sum_{i=1}^{p} y_i^{'} y_i\right) - \left(\sum_{i=1}^{p} y_i^{'}\right)\left(\sum_{i=1}^{p} y_i\right)}{\sqrt{\left(p\sum_{i=1}^{p} y_i^2 - \left(\sum_{i=1}^{p} y_i\right)^2\right)\left(\left(p\sum_{i=1}^{p} y_i^{'2} - \left(\sum_{i=1}^{p} y_i^{'}\right)^2\right)\right)}} \tag{10}$$

$$NSE = 1 - \frac{\sum_{i=1}^{p}\left(y_i - y_i^{'}\right)^2}{\sum_{i=1}^{p}(y_i - \bar{y})^2} \tag{11}$$

TABLE 6. MAE, RMSE, r and NSE values over each month alone.

Model		ACCESS-S1				MLP				CNN			
Month	Average	MAE	RMSE	r	NSE	MAE	RMSE	r	NSE	MAE	RMSE	r	NSE
January	541.097	206.314	265.040	0.408	0.104	214.238	252.245	0.655	0.188	194.035	**230.333**	0.701	0.323
February	631.415	239.770	279.983	0.495	0.231	197.998	232.718	0.732	0.469	171.258	**202.810**	0.796	0.596
March	668.402	301.783	339.437	0.414	0.147	149.446	159.975	0.935	0.810	118.637	**140.079**	0.939	0.855
April	449.361	132.317	**156.818**	0.355	-0.009	146.714	178.471	-0.396	-0.306	129.837	162.478	-0.040	-0.083
May	315.475	121.923	142.910	0.133	-0.234	102.837	120.743	0.363	0.119	105.862	**120.375**	0.604	0.125
June	190.843	77.442	**100.907**	0.111	-0.103	98.311	112.734	-0.185	-0.377	99.572	116.021	0.134	-0.459
July	131.661	80.658	100.985	0.183	-0.331	67.039	**83.818**	0.618	0.083	78.206	89.043	0.132	-0.035
August	108.350	42.783	**59.126**	0.710	0.479	50.489	76.719	0.731	0.124	63.889	81.699	0.300	0.006
September	88.324	45.895	**68.216**	0.687	0.388	72.634	90.623	0.105	-0.081	75.944	83.381	0.682	0.085
October	87.512	75.405	98.584	0.702	0.435	88.089	122.037	0.562	0.135	83.143	**93.379**	0.758	0.493
November	151.814	92.632	**130.358**	0.816	0.497	130.388	175.460	0.361	0.089	121.842	142.098	0.765	0.403
December	253.325	123.221	**144.844**	0.531	0.201	129.904	174.727	0.455	-0.162	134.601	154.862	0.357	0.087

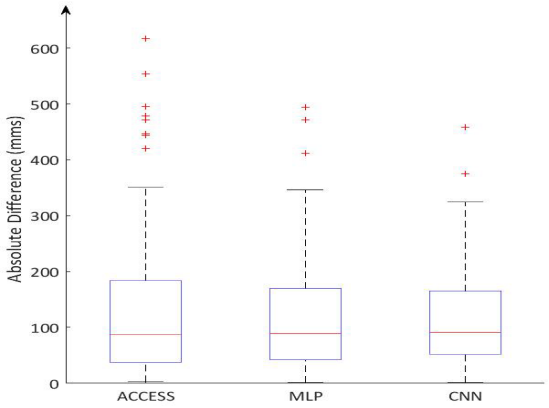

FIGURE 4. Boxplots of the absolute difference between actual and forecasted values for each prediction model.

17.4999° S, 146.2500° E. In addition, a MLP was designed. The Keras package was used to implement the MLP. The same training, validation and testing portions were selected with the MLP. The number of hidden neurons and activation functions were selected using trial and error method. A MLP consisting of 23 neurons in the hidden layer revealed the best prediction accuracy. Hyperbolic tangent transfer functions were selected between the input-hidden and hidden-output layers.

MAE, RMSE, r and NSE for each model over the hold-out sample are shown in TABLE 5. Better MAE, RMSE, r and NSE values were obtained with the generated CNN. A 10.691 mm difference was recorded in terms of MAE and 37.006 in terms of RMSE compared to ACCESS-S1. A 6.020 mm difference was obtained in terms of MAE and 15.941 in terms of RMSE compared to MLP. Higher r and NSE values were also revealed with the generated CNN compared to ACCESS-S1 and MLP. This reveals the appropriateness of this kind of methods in predicting rainfall values.

To investigate further, the hold-out sample was divided into 12 datasets to analyse the performance of each model on monthly basis. Hereafter, the statistical measurements were calculated over each month alone between January 2001 and December 2012. MAE, RMSE, r and NSE values for each month using each model are shown in TABLE 6. The second column represents the annual average of each month based on the training and validation datasets.

The CNN revealed better prediction accuracies in terms of RMSE in 6 out of the 12 months compared to ACCESS-S1: January, February, March, May, July and October. ACCESS-S1 showed better performance in six months: April, June, August, September, November and December.

The CNN showed better performance in 9 out of the 12 months compared to the MLP: January, February, March, April, May, September, October, November and December. The MLP showed better accuracy in the remaining three months.

The first three months of the year held the highest rainfall averages. The CNN model revealed better accuracy in all of the three months. This demonstrates the ability of the CNN in performing with high ranges. ACCESS-S1 performance was better in two out of the three months with lowest annual averages (September, October). In 10 out of the 12 months, the CNN was found to be the best or the second-best prediction model. Negative NSE values were obtained for April, June and July. The STDs of the three months over the predicted values were low compared to the STDs over the actual values (April – actual: 156.147, predictions: 32.010; June – actual: 96.064, predictions: 47.244; July – actual: 87.535, predictions: 31.019). The spread of data over actual values was at least two times higher than predicted values in each of the three months. This could be related to the CNN not being able to learn the non-linear relationships for those three months. Additional reason could be the need for extra information (data) that may affect the rainfall variability over the three months and was not found in generated dataset.

Figure 4 represents the absolute difference between actual and forecasted values for each prediction model. The number outliers in the ACCESS-S1 model was higher than the number of outliers in the CNN model. This can be referred to the high RMSE obtained with the ACCESS-S1 model. It is

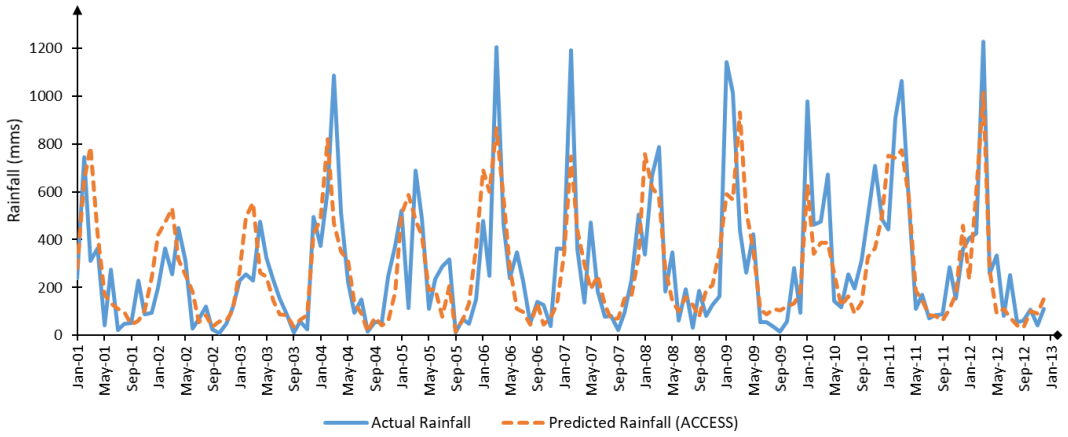

FIGURE 5. Rainfall values against ACCESS output between Jan-2001 and Dec-2012.

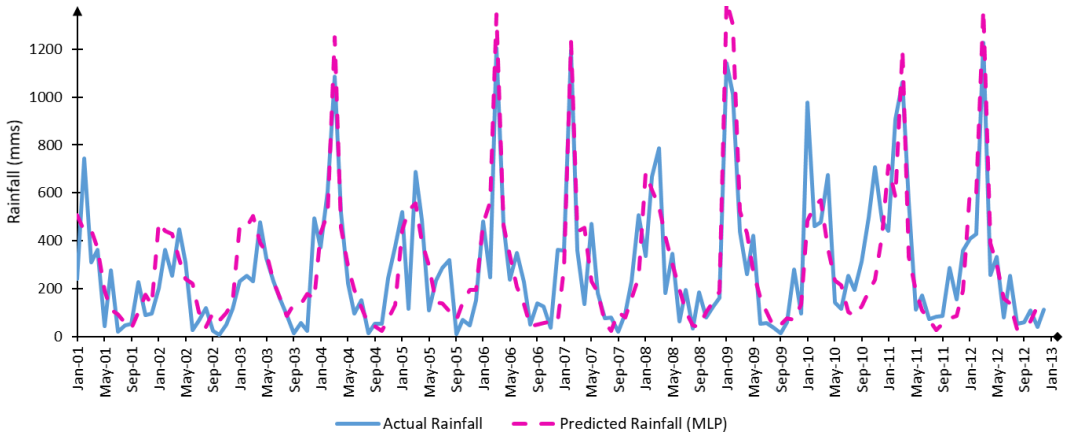

FIGURE 6. Rainfall values against MLP output between Jan-2001 and Dec-2012.

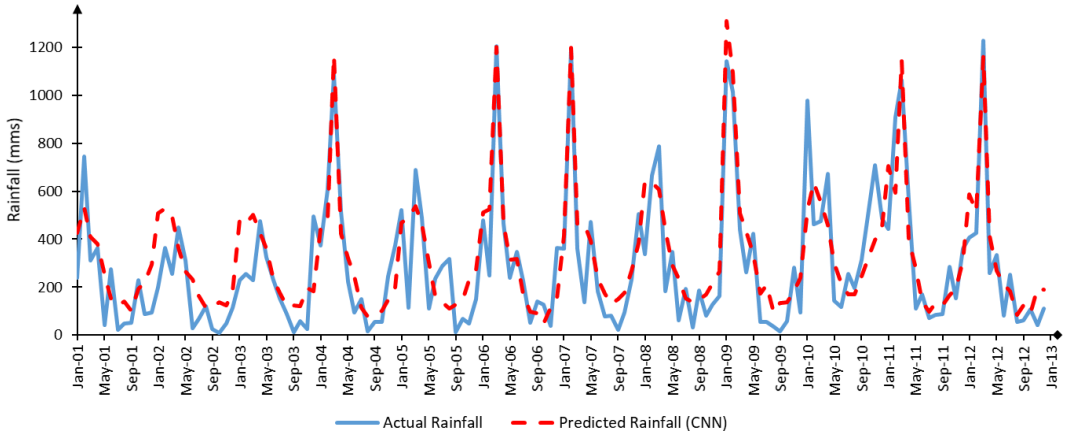

FIGURE 7. Rainfall values against CNN output between Jan-2001 and Dec-2012.

clearly demonstrated that the machine learning models were performing better with months with high annual averages, while ACCESS-S1 revealed better accuracy with months with lower annual averages. Therefore, as a part of future work, ensemble techniques could be investigated to combine the efficiencies of each prediction model.

Figure. 5 represents the actual rainfall values compared to ACCESS-S1 output between Jan-2001 and Dec-2012. Figure. 6 represents the actual rainfall values compared to MLP output between Jan-2001 and Dec-2012. Figure. 7 represents the actual rainfall values compared to CNN output between Jan-2001 and Dec-2012. The continuous line in each

figure shows actual rainfall, while the dotted line shows the output of each model (ACCESS-S1, MLP and CNN) respectively. It is clearly shown that the generated machine learning based models performed better than ACCESS-S1 in rainy months with better performance with the CNN model. In most of the peaks, rainfall was predicted accurately. On the other hand, the rainfall values were not well predicted in months with low annual averages, as shown in September 2002, September 2003, September 2004 etc. ACCESS-S1 held a better ability in predicting rainfall values for months with low annual averages. In most of the instances that represent months with low rainfall averages, the generated CNN anticipated higher rainfall values compared to actual rainfall.

VI. CONCLUSION

In this paper, a neural network based approach has been proposed to forecast monthly rainfall for a selected location in Australia. A deep CNN was developed to predict monthly rainfall. The developed model was compared to the first version of the Australian Community Climate and Earth-System Simulator (ACCESS-S1) and a Multi-Layered Perceptron (MLP), where better performance was revealed with the proposed CNN model. This study highlighted the ability of CNN in performing in a rainfall prediction task. In addition, the capability of convolutional neural networks in performing in a monthly rainfall prediction has been evaluated. The study showed that the CNN was performing better in months with high annual averages compared to alternative approaches. In future research, ensemble techniques will be incorporated to combine the diversities of the models. Also, additional locations and datasets will be incorporated. The architecture of the network model will be examined further to enhance the accuracy of predictions.

REFERENCES

[1] D. Hudson, A. G. Marshall, Y. Yin, O. Alves, and H. H. Hendon, "Improving intraseasonal prediction with a new ensemble generation strategy," *Monthly Weather Rev.*, vol. 141, pp. 4429–4449, Dec. 2013.
[2] W. Drosdowsky and M. C. Wheeler, "Predicting the onset of the north Australian wet season with the POAMA dynamical prediction system," *Weather Forecasting*, vol. 29, no. 1, pp. 150–161, 2014.
[3] A. Cottrill *et al.*, "Seasonal climate prediction in the Pacific using the POAMA coupled model forecast system," CACWCR, Australia, Tech. Rep. 048, 2012.
[4] D. Hudson *et al.*, "ACCESS-S1: The new bureau of meteorology multi-week to seasonal prediction system," *J. Southern Hemisphere Earth Syst. Sci.*, vol. 67, no. 3, pp. 132–159, 2017.
[5] Z. M. Yaseen, A. El-Shafie, O. Jaafar, H. A. Afan, and K. N. Sayl, "Artificial intelligence based models for stream-flow forecasting: 2000–2015," *J. Hydrol.*, vol. 530, pp. 829–844, Nov. 2015.
[6] C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, "Predicting stock returns by classifier ensembles," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2452–2459, 2011.
[7] L. Zhang, B. Verma, and D. Stockwell, "Class-semantic color-texture textons for vegetation classification," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, 2015, pp. 354–362.
[8] S. S. Talathi, "Hyper-parameter optimization of deep convolutional networks for object recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3982–3986.
[9] M. A. Ahmadi, R. Soleimani, M. Lee, T. Kashiwao, and A. Bahadori, "Determination of oil well production performance using artificial neural network (ANN) linked to the particle swarm optimization (PSO) tool," *Petroleum*, vol. 1, no. 2, pp. 118–132, 2015.
[10] K. Muralitharan, R. Sakthivel, and R. Vishnuvarthan, "Neural network based optimization approach for energy demand prediction in smart grid," *Neurocomputing*, vol. 273, pp. 199–208, Jan. 2018.
[11] N. D. Truong *et al.*, "Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram," *Neural Netw.*, vol. 105, pp. 104–111, Sep. 2018.
[12] R. Haidar, I. Koprinska, and B. Jeffries, "Sleep apnea event detection from nasal airflow using convolutional neural networks," in *Proc. 24th Int. Conf. Neural Inf. Process. (ICONIP)*, Guangzhou, China, 2017, pp. 819–827.
[13] P. M. Leod, B. Verma, and M. Zhang, "Optimizing configuration of neural ensemble network for breast cancer diagnosis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2014, pp. 1087–1092.
[14] W. W. Guo, L. D. Li, and G. Whymark, "Simulating wheat yield in New South Wales of Australia using interpolation and neural networks," in *Proc. 17th Int. Conf. Neural Inf. Process.*, Sydney, NSW, Australia, 2010, pp. 708–715.
[15] F. Mekanik and M. A. Imteaz, "Capability of artificial neural networks for predicting long-term seasonal rainfalls in east Australia," in *Proc. 20th Int. Congr. Modelling Simulation (Modsim)*, 2013, pp. 2674–2680.
[16] S. Chaudhuri and S. Chattopadhyay, "Neuro-computing based short range prediction of some meteorological parameters during the pre-monsoon season," *Soft Comput.*, vol. 9, no. 5, pp. 349–354, 2005.
[17] X. Gan, L. Chen, D. Yang, and G. Liu, "The research of rainfall prediction models based on MATLAB neural network," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Sep. 2011, pp. 45–48.
[18] P. T. Nastos, K. P. Moustris, I. K. Larissi, and A. G. Paliatsos, "Rain intensity forecast using artificial neural networks in Athens, Greece," *Atmos. Res.*, vol. 119, pp. 153–160, Jan. 2013.
[19] K. G. Sheela and S. N. Deepa, "Performance analysis of modeling framework for prediction in wind farms employing artificial neural networks," *Soft Comput.*, vol. 18, no. 3, pp. 607–615, 2014.
[20] H. Vathsala and S. G. Koolagudi, "Prediction model for peninsular Indian summer monsoon rainfall using data mining and statistical approaches," *Comput. Geosci.*, vol. 98, pp. 55–63, Jan. 2017.
[21] S. Hardwinarto and M. Aipassa, "Rainfall monthly prediction based on artificial neural network: A case study in Tenggarong Station, East Kalimantan-Indonesia," *Procedia Comput. Sci.*, vol. 59, pp. 142–151, Jan. 2015.
[22] L. Ayalew, D. P. F. Möller, and G. Reik, "Using artificial neural networks (ANN) for real time flood forecasting, the Omo river case in southern Ethiopia," in *Proc. Summer Comput. Simulation Conf.*, San Diego, CA, USA, 2007, pp. 1–7.
[23] R. C. Deo and M. Şahin, "Application of the artificial neural network model for prediction of monthly standardized precipitation and evapotranspiration index using hydrometeorological parameters and climate indices in eastern Australia," *Atmos. Res.*, vols. 161–162, pp. 65–81, Jul./Aug. 2015.
[24] J. A. Le, H. M. El-Askary, M. Allali, and D. C. Struppa, "Application of recurrent neural networks for drought projections in California," *Atmos. Res.*, vol. 188, pp. 100–106, May 2017.
[25] S. Campisi-Pinto, J. Adamowski, and G. Oron, "Forecasting urban water demand via wavelet-denoising and neural network models. Case study: City of Syracuse, Italy," *Water Resour. Manage.*, vol. 26, no. 12, pp. 3539–3558, 2012.
[26] A. Elzwayie, A. El-Shafie, Z. M. Yaseen, H. A. Afan, and M. F. Allawi, "RBFNN-based model for heavy metal prediction for different climatic and pollution conditions," *Neural Comput. Appl.*, vol. 28, no. 8, pp. 1991–2003, 2017.
[27] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks," in *Artificial Neural Networks*, vol. 458. New York, NY, USA: Humana Press, 2008, pp. 14–22.
[28] K. C. Luk, J. E. Ball, and A. Sharma, "An application of artificial neural networks for rainfall forecasting," *Math. Comput. Model.*, vol. 33, nos. 6–7, pp. 683–693, 2001.
[29] M. S. Baawain, M. H. Nour, A. G. El-Din, and M. G. El-Din, "El Niño southern-oscillation prediction using southern oscillation index and $Niño_3$ as onset indicators: Application of artificial neural networks," *J. Environ. Eng. Sci.*, vol. 4, no. 2, pp. 113–121, 2005.
[30] S. Chattopadhyay, "Feed forward artificial neural network model to predict the average summer-monsoon rainfall in India," *Acta Geophys.*, vol. 55, no. 3, pp. 369–382, 2007.
[31] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand," *Hydrol. Earth Syst. Sci.*, vol. 13, no. 8, pp. 1413–1425, 2009.

[32] H. R. K. Nagahamulla, U. R. Ratnayake, and A. Ratnaweera, "Monsoon rainfall forecasting in Sri Lanka using artificial neural networks," in *Proc. 6th Int. Conf. Ind. Inf. Syst.*, 2011, pp. 305–309.

[33] F. Mekanik, M. A. Imteaz, S. Gato-Trinidad, and A. Elmahdi, "Multiple regression and artificial neural network for long-term rainfall forecasting using large scale climate modes," *J. Hydrol.*, vol. 503, pp. 11–21, Oct. 2013.

[34] T. Kashiwao, K. Nakayama, S. Ando, K. Ikeda, M. Lee, and A. Bahadori, "A neural network-based local rainfall prediction system using meteorological data on the Internet: A case study using data from the Japan Meteorological Agency," *Appl. Soft Comput.*, vol. 56, pp. 317–330, Jul. 2017.

[35] S. Karmakar, M. K. Kowar, and P. Guhathakurta, "Long-range monsoon rainfall pattern recognition and prediction for the subdivision 'EPMB' chhattisgarh using deterministic and probabilistic neural network," in *Proc. 7th Int. Conf. Adv. Pattern Recognit.*, 2009, pp. 367–370.

[36] Z.-L. Wang and H.-H. Sheng, "Rainfall prediction using generalized regression neural network: Case study Zhengzhou," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, 2010, pp. 1265–1268.

[37] K. P. Moustris, I. K. Larissi, P. T. Nastos, and A. G. Paliatsos, "Precipitation forecast using artificial neural networks in specific regions of Greece," *Water Resour. Manage.*, vol. 25, no. 8, pp. 1979–1993, 2011.

[38] N. A. Charaniya and S. V. Dudul, "Focused time delay neural network model for rainfall prediction using Indian Ocean Dipole index," in *Proc. 4th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, 2012, pp. 851–855.

[39] S. Khedhiri, "Artificial neural network for forecasting rainfall pattern in Prince Edward Island, Canada," *Int. J. Environ. Stud.*, vol. 72, no. 2, pp. 331–340, 2015.

[40] L. Jaedong and L. Jee-Hyong, "Constructing efficient regional hazardous weather prediction models through big data analysis," *Int. J. Fuzzy Logic Intell. Syst.*, vol. 16, no. 1, pp. 1–12, 2016.

[41] F. Mekanik, M. A. Imteaz, and A. Talei, "Seasonal rainfall forecasting by adaptive network-based fuzzy inference system (ANFIS) using large scale climate signals," *Climate Dyn.*, vol. 46, nos. 9–10, pp. 3097–3111, 2016.

[42] A. Kusiak, X. Wei, A. P. Verma, and E. Roz, "Modeling and prediction of rainfall using radar reflectivity data: A data-mining approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2337–2342, Apr. 2013.

[43] R. C. Deo and M. Şahin, "Application of the extreme learning machine algorithm for the prediction of monthly effective drought index in eastern Australia," *Atmos. Res.*, vol. 153, pp. 512–525, Feb. 2015.

[44] A. M. Bagirov and A. Mahmood, "A comparative assessment of models to predict monthly rainfall in Australia," *Water Resour. Manage.*, vol. 32, no. 5, pp. 1777–1794, 2018.

[45] Z. M. Yaseen *et al.*, "Rainfall pattern forecasting using novel hybrid intelligent model based ANFIS-FFA," *Water Resour. Manage.*, vol. 32, pp. 105–122, Jan. 2018.

[46] M. A. Sojitra, R. C. Purohit, and P. A. Pandya, "Comparative study of daily rainfall forecasting models using adaptive-neuro fuzzy inference system (ANFIS)," *Current World Environ.*, vol. 10, no. 2, pp. 529–536, 2015.

[47] Z. M. Yaseen *et al.*, "Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq," *J. Hydrol.*, vol. 542, pp. 603–614, Nov. 2016.

[48] F. Fahimi, Z. M. Yaseen, and A. El-Shafie, "Application of soft computing based hybrid models in hydrological variables modeling: A comprehensive review," *Theor. Appl. Climatol.*, vol. 128, pp. 875–903, 2017.

[49] N. S. Raghavendra and P. C. Deka, "Support vector machine applications in the field of hydrology: A review," *Appl. Soft Comput.*, vol. 19, pp. 372–386, Jun. 2014.

[50] V. Nourani, A. H. Baghanam, J. Adamowski, and O. Kisi, "Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review," *J. Hydrol.*, vol. 514, pp. 358–377, Jun. 2014.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[52] F. Shaheen and B. Verma, "An ensemble of deep learning architectures for automatic feature extraction," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–5.

[53] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. IJCAI*, 2015, pp. 3995–4001.

[54] W. Sun, T. B. Tseng, J. Zhang, and W. Qian, "Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data," *Comput. Med. Imag. Graph.*, vol. 57, pp. 4–9, Apr. 2017.

[55] C. Liu, W. Hou, and D. Liu, "Foreign exchange rates forecasting with convolutional neural network," *Neural Process. Lett.*, vol. 46, no. 3, pp. 1095–1119, 2017.

[56] D. Bardou, K. Zhang, and S. M. Ahmad, "Classification of breast cancer based on histology images using convolutional neural networks," *IEEE Access*, vol. 6, pp. 24680–24693, 2018.

[57] S. McCloskey, R. Haidar, I. Koprinska, and B. Jeffries, "Detecting hypopnea and obstructive apnea events using convolutional neural networks on wavelet spectrograms of nasal airflow," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining (PAKDD)*, Melbourne, VIC, Australia, 2018, pp. 361–372.

[58] *Bureau of Meteorology. Climate Data*. Accessed: Apr. 4, 2016. [Online]. Available: http://www.bom.gov.au/climate/data/

[59] A. Haidar and B. Verma, "Monthly rainfall categorization based on optimized features and neural network," in *Proc. 30th Australas. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, 2017, pp. 208–220.

[60] *Royal Netherlands Meteorological Institute Climate Explorer*. Accessed: Aug. 4, 2015. [Online]. Available: https://climexp.knmi.nl/start.cgi

[61] C. K. Folland, J. Shukla, J. Kinter, and M. J. Rodwell. (2002). *C20C: The Climate of the Twentieth Century Project*. [Online]. Available: http://cola.gmu.edu/c20c/

[62] Earth System Research Laboratory. Accessed: Aug. 15, 2015. [Online]. Available: https://www.esrl.noaa.gov/

[63] *Solar Influences Data Analysis Center*. Accessed: Aug. 4, 2015. [Online]. Available: http://sidc.oma.be/

[64] D. N. Moriasi, J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith, "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations," *Trans. ASABE*, vol. 50, no. 3, pp. 885–900, 2007.

[65] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[66] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, 2016, pp. 265–283.

[67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[68] *National Computational Infrastructure*. Accessed: Mar. 24, 2018. [Online]. Available: https://geonetwork.nci.org.au/geonetwork/srv/eng/catalog.search#/metadata/f5023_1291_6579_1882

ALI HAIDAR received the M.S. degree from The University of Sydney, Australia. He is currently pursuing the Ph.D. degree with the School of Engineering and Technology, Central Queensland University, Sydney, Australia. His current research interests include machine learning, deep learning, and computational intelligence.

BRIJESH VERMA is currently a Professor and the Director of the Centre for Intelligent Systems (CIS), School of Engineering and Technology, Central Queensland University, Brisbane, Australia. He has authored/co-authored/co-edited 13 books, including *Roadside Video Data Analysis: Deep Learning*, nine book chapters, and over 150 papers in areas, such as neural networks, deep learning, evolutionary algorithms, pattern recognition, computer vision, image processing, digital mammography, and Web information retrieval. His main research interests include computational intelligence and pattern recognition.

He is the President of INNS (International Neural Network Society) Australia Chapter and a Member of the Australian Research Council College of Experts. He was the Chair of the IEEE Computational Intelligence Society's Queensland Chapter and under his leadership the Chapter won Outstanding Chapter Award from IEEE CIS. He is an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, the Editor-in-Chief of the *International Journal of Computational Intelligence and Applications*, and an Editorial Board Member of a number of international journals, including *Neural Computing and Applications*.