Precipitation forecast using RNN variants by analyzing Optimizers and Hyperparameters for Time-series based Climatological Data

Original Scientific Paper

J. Subha
Centre for Information Technology and Engineering,
Manonmaniam Sundaranar University, Tirunelveli – 12, Tamil Nadu, India
subha.arhip@gmail.com

S. Saudia
Centre for Information Technology and Engineering,
Manonmaniam Sundaranar University, Tirunelveli – 12, Tamil Nadu, India
saudiasubash@msuniv.ac.in

Abstract *– Flood is a significant problem in many regions of the world for the catastrophic damage it causes to both property and human lives; excessive precipitation being the major cause. The AI technologies, Deep Learning Neural Networks and Machine Learning algorithms attempt realistic solutions to numerous disaster management challenges. This paper works on RNN-based rainfall/ precipitation forecasting models by investigating the performances of various Recurrent Neural Network (RNN) architectures, Bidirectional RNN (BRNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU using NASAPOWER datasets of Andhra Pradesh (AP) and Tamil Nadu (TN) in India. The different stages in the workflow of the methodology are Data collection, Data pre-processing, Data splitting, Defining hyperparameters, Model building and Performance evaluation. Experiments for identifying improved optimizers and hyperparameters for the time-series climatological data are investigated for accurate precipitation forecast. The metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and Root Mean Squared Logarithmic Error (RMSLE) values are used to compare the precipitation predictions of different models. The RNN variants and ensemble models, BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU produce predictions with RMSLE values of 2.448, 0.555, 0.255, 1.305, 1.383, 0.364, 1.740 for AP and 1.735, 0.663, 0.152, 0.889, 1.118, 0.379, 1.328 for TN respectively. The best performing RNN model, GRU when ensembled with the existing statistical model SARIMA produces an RMSLE value of 0.754 and 1.677 respectively for AP and TN.*

Keywords*: deep Learning, optimizers, hyperparameters, RNN, BRNN, LSTM, GRU, SARIMA*

Received: November 1, 2023; Received in revised form: February 9, 2024; Accepted: February, 12 2024

1. INTRODUCTION

Around the world, floods regularly cause enormous losses; India also suffers the most serious damages [1, 2]. The prime cause being excessive precipitation [3, 4] that occurs suddenly resulting in hazardous flood conditions and difficulties for the people. Out of all natural disasters, floods in India account for 56% of all fatalities [4]. According to Central Water Commission (CWC) data for India, between 1953 and 2020 there was an average of 1676 flood-related fatalities every year. Table 1 depicts the Flood-affected areas and flood damages in India during the period from 1953 to 2020 [5].

In the statement on the climate of India released by the India Meteorological Department (IMD) in the year 2022 [6] concerning the data from 1971 to 2020, the majority of India had a high long-period average (LPA) precipitation of 108%. During the years, South Peninsular India received seasonal monsoon precipitation equal to 122% LPA; Central India and Northwest India received seasonal precipitation equal to 119% and 101% LPAs respectively; and East & Northeast India received seasonal monsoon precipitation equal to 82% LPA [7]. The heavy rainfall days showed significantly increasing flood trends over peninsular India [8]. The impact of excessive rainfall is the cause of flood frequency in various parts of the world especially India, Indonesia and Spain [8-11]. Also, from [9] and [11], the amount of daily rainfall turned out to have a stronger correlation with floods. This thought motivates the research to make flood forecasts for safe living in extreme rainfall-receiving areas of India. Rainfall events are classified

as moderate when it is between 2.5 and 64.5 mm/day, while events that fall beyond 64.5 mm/day are classified as heavy rainfall [8]-[10]. So, when the rainfall is over 64.5 mm/day, there are chances of flooding. Thus, this research which involved rainfall prediction is important to forecast flood risk based on predicted rainfall values greater than 64.5 mm/day.

Flooding is possible in heavy precipitation-receiving regions like South Peninsular India or South India. The union territories of the Andaman and Nicobar Islands, Lakshadweep and Puducherry are included in South India in addition to the Indian states of Andhra Pradesh, Karnataka, Kerala, Tamil Nadu and Telangana. South India encompasses 20% of the nation's population and 19.31% of its total area (635,780 km2 or 245,480 sq. miles) [12]. It is well recognized that anticipating flood disasters requires an awareness and analysis of variations in precipitation [13]. The purpose of this research is to analyze time-series climatological data and execute RNN-based precipitation forecast models to prevent flooding in two of the states in South Peninsular India: Andhra Pradesh (AP) and Tamil Nadu (TN). The daily precipitation values of TN and AP for the last 20 years from 2002 to 2022 as identified from the NASAPOWER dataset [14], range from 0 to 178.98 mm/day.

Table 1. Flood damages in India from1953 to 2020

Description of Flood Damages	Measure Unit	Total damage	Average damage
Area Affected	million hectares (m.ha.)	493	7.24
Population Affected	million	2199	32.34
Crops Area Affected	m. ha.	276	4.06
Values of Crops Affected	Rs. Crore	131462	1933.27
Number of Houses Damaged	Nos.	82525198	1213606
Value of Damaged Houses	Rs. Crore	57018	838.50
Number of Cattle Lost	Nos.	6182943	90926.00
Number of Human Lives Lost	Nos.	113943	1676.00
Value of Damaged Public Utilities	Rs. Crore	234149	3443.37
Value of Total Damages (Inc. Houses, Crops, public utilities)	Rs. Crore	437150	6428.67

Presently, many technologies, including Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) provide solutions for different domains of disaster management. ML algorithms, such as Regression [15], Support Vector Machine (SVM) [7], Decision Trees (DT) [16, 17], Naive Bayes [18] and K-Nearest Neighbors (KNN) [19] have been effectively used to construct precipitation prediction or classification models in a variety of domains. DL algorithms, such as Artificial Neural Networks (ANN) [20], Recurrent Neural Networks (RNN) [21, 22], Convolutional Neural Networks (CNN) [23] and Generative Adversarial Networks (GAN) [24] play an essential role in processing and analyzing massive amounts of precipitation data to deliver meaningful information. Venkatesh, et al. (2021) [25] constructed a precipitation prediction system using GAN with a CNN upon time-series annual precipitation data of 36 subdivisions in India from 1901 to 2015. This provided a better prediction for summer, winter, pre-monsoon, and post-monsoon precipitations with an accuracy of 99%. S Aswin, et al. (2018) [26] developed a model for predicting precipitation using DL architectures, LSTM and CNN for the Global Precipitation Climatology Project (GPCP) upon a monthly precipitation dataset that encompasses the time frame of July 1979 to January 2018. Also, it declared that LSTM and CNN produced RMSE of 2.55 and 2.44 respectively.

Haq, et al. [27] constructed an LSTM model based on El-Nino and Indian Ocean Dipole (IOD) precipitation data to predict precipitation for Sidoarjo, East Java and Indonesia. With a hidden layer, batch size and drop period of 100, 32 and 150 respectively, the model performed with a Mean Arctangent Absolute Percentage Error (MAAPE) value of 0.5810. Dechao et al. [28] created a DL-based forecast on the intensity of the regional precipitation in the next two hours with the use of 3 dimensional CNN (Conv3D), GRU algorithms and radar images. Ouma, et al. [29] presented precipitation and time-series trend analysis using LSTM and Wavelet Neural Networks (WNN). Moreover, predictions were made using hydrologic basin precipitation streamflow data and satellite-based meteorological data from 1980–2009. Both models performed well and predicted the precipitation with R2 values of 0.8610 and 0.7825 respectively. Samad et al. [30] built a model for rainfall prediction based on an Australian dataset for the regions of Albany, Walpole, and Witchcliffe. The LSTM network outperformed the ANN after comparison using different performance measures including MSE, RMSE and MAE. The LSTM model accurately predicted the precipitation with RMSE values of 5.343, 6.280, and 7.706 for the three regions respectively. Dada et al. [21] proposed four Neural Network models: Feed Forward Neural Network (FFNN), RNN, Elman Neural Network (ENN) and Cascade Forward Neural Network (CFNN) for predicting precipitation using India's precipitation data from the Kaggle repository. Additionally, it is clear from the statistical findings that the ENN model outperformed the other three models. ENN was discovered to have the best performance with the lowest RMSE, MSE, and MAE values of 6.360, 40.45, and 0.54 respectively. Saha et al. [31] used an ensemble regression tree model utilizing data from 1948 to 2015 for estimating monsoon precipitation over homogeneous regions of India. Forecast errors for the monsoons in central, northeast, north-west and south-peninsular India are 4.1%, 5.1%, 5.5% and 6.4% respectively.

From the baseline references [1-5], it is found necessary to build an effective flood warning system to prevent recurring harm sustained by people. Also, pre-

cipitation is the most significant reason for generating floods. Time-series data-based precipitation forecast can aid in the decision-making processes associated with flood and disaster to manage, flood, control floods and plan safety preparations [31, 32].

Also, based on a survey of related work [21, 25-31], it has been determined that articles for predicting precipitation use DL neural network models such as LSTM, ANN, GRU, FFNN, RNN, ENN, and CFNN. Also, measures such as RMSE and MAE are used to assess the expected precipitation. Thus, to assist in the decision-making procedures for preventing floods which demand extremely precise predicted precipitation, the study decides to carry out experiments and a thorough analysis by experimenting with a variety of optimizers and hyperparameters to forecast precipitation using RNN variants on time-series precipitation data. This study focuses on creating RNN variant models using BRNN, LSTM, GRU and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU to improve the effectiveness of RNN-based precipitation forecast. Finally, the paper also attempts an ensemble model for predicting precipitation using the lowest-error models of RNN variants with the statistical model, SARIMA. The most effective optimizer and hyperparameters as identified from the experiments are selected to predict precipitation using the RNN variant model. Also, the results of the proposed RNN variants and ensemble models are compared in terms of error metrics, with those from publications by S. Aswin et al. [26], Haq et al. [27], Ouma et al. [29], Samad A, et al. [30], Dada et al. [21], and Saha et al. [31]. The comparative analysis demonstrates that the proposed model outperforms all other models and techniques under comparison. The rest of this paper is organized in three more sections: Section 2 describes the workflow of the proposed methodology; Section 3 presents the results of the experiments and Section 4 summarizes the findings of the proposed work.

2. PROPOSED SYSTEM

The proposed methodology for designing the models for daily precipitation forecasting for the southern states of AP and TN is shown in Fig. 1. The proposed system predicts precipitation for AP and TN using RNN variants such as BRNN, LSTM, GRU, and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU with suitable hyperparameters and optimizers. The different stages in the workflow of the methodology are Data collection, Data pre-processing, Data splitting, Defining hyperparameters, Model building and Performance evaluation. Time-series-based precipitation data from 2002 to 2022 is downloaded from the NASAPOWER website- https://power.larc.nasa.gov/ [14].

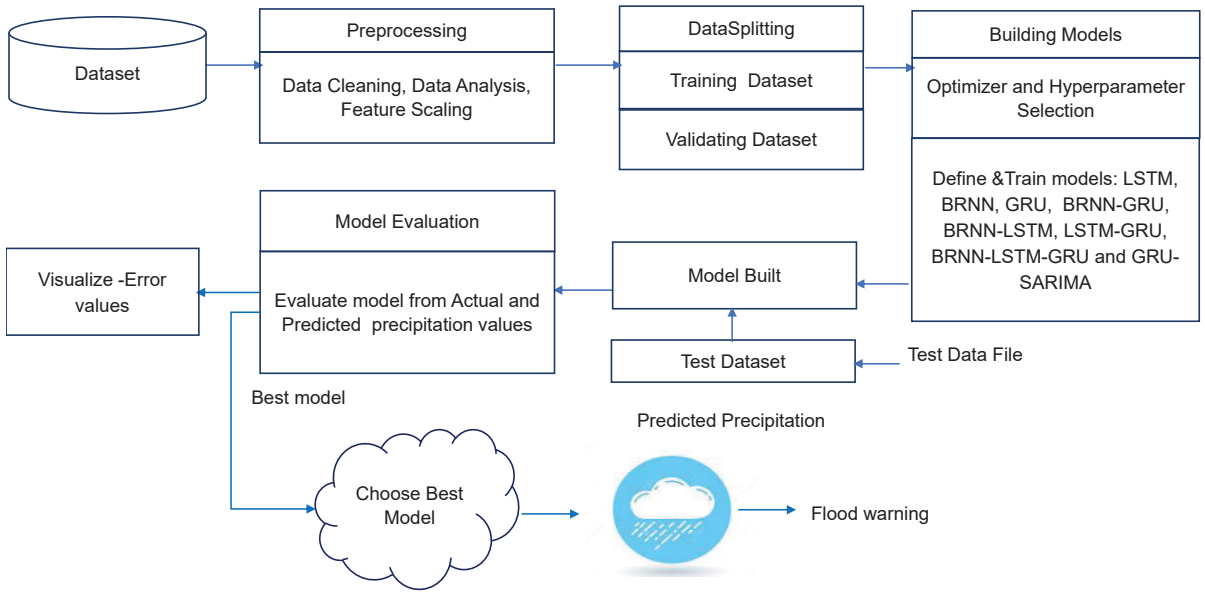


Fig. 1. Methodology of the proposed RNN based precipitation forecast system

The best hyperparameters and optimizers for the RNN variants, BRNN, LSTM and GRU are identified and used to build the precipitation forecast models. Several ensemble models like BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU are also experimented with for predicting precipitation. The performances of different RNN and the ensemble models are evaluated using MSE, RMSE, MAE and RMSLE. Finally, a GRU-SARIMA is created with the lowest error-producing GRU and the existing statistical model SARIMA for predicting precipitation.

2.1. DATASET

The time-series data is obtained from the website https://power.larc.nasa.gov/ [14] for training the DL and ensemble algorithms. This website provides climatological data obtained from satellite observations of agricultural and renewable energy usage. The time-series climatological dataset is collected for the southern Indian states of AP and TN in Comma-Separated Value [CSV] file format for the years from 2002 to 2022. This

time-series dataset had 7670 records. In this study, only the precipitation data and the corresponding date of year are taken from the dataset. The date of year feature is created using the to_datetime () function in the pandas package of Python. This forms a univariate time series of precipitation data covering 20 years from 2002 to 2022. Utilizing the Python packages pandas and matplotlib, the dataset is visualized as in Fig. 2 which shows the daily, weekly and monthly precipitation totals of AP and TN states for the years 2002–2022. Fig. 2 depicts the fluctuation in precipitation and the peak points corresponding to the highest and lowest precipitation.

Every year from June to December, a time series pattern in the form of increasing trends in precipitation is seen in the TN and AP statistics.

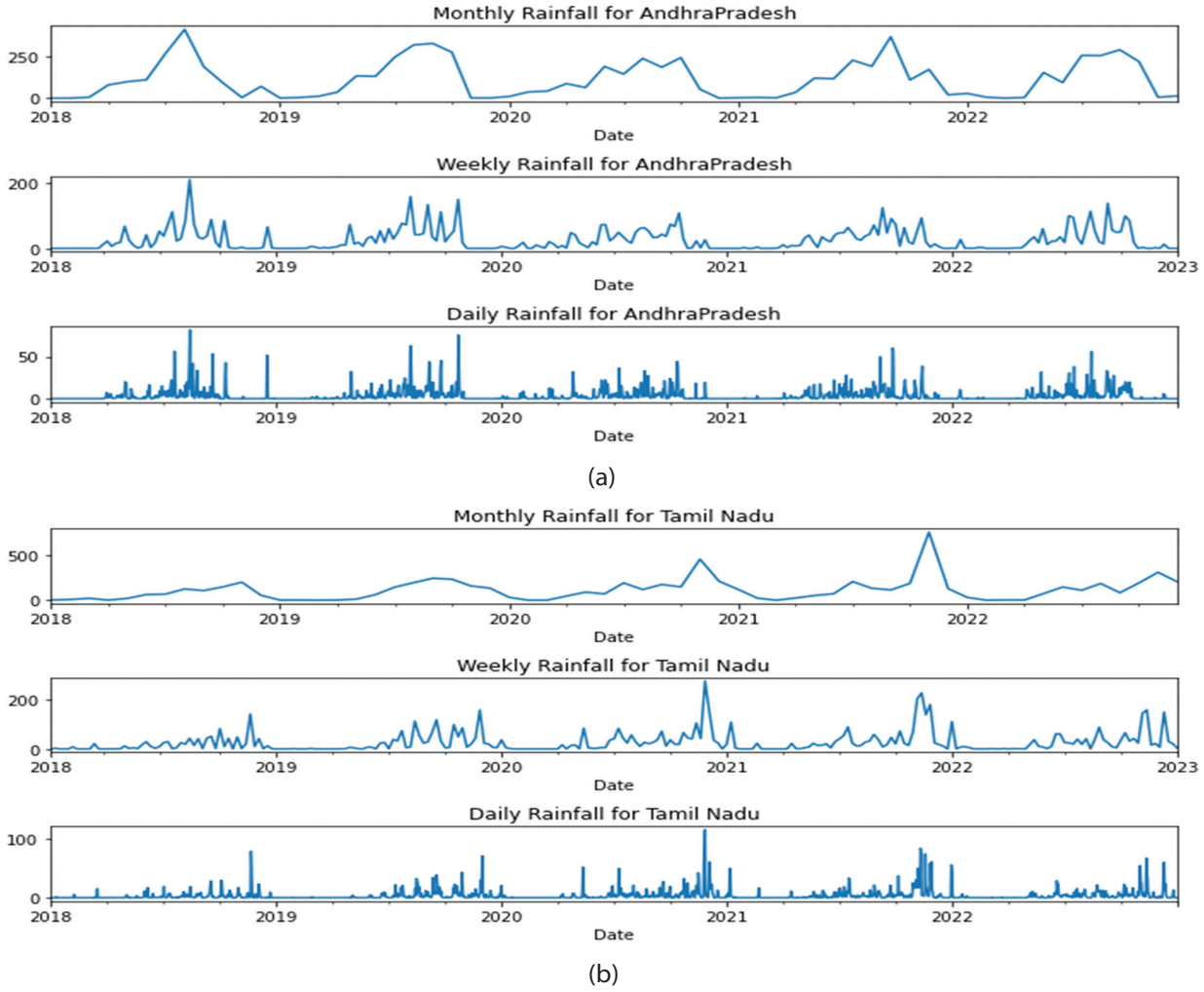


Fig. 2. Line Plots showing increasing trends in the daily, weekly, and monthly precipitation statistics of a) Andhra Pradesh b) Tamil Nadu from 2018 to 2022

Fig. 2 illustrates the seasonal pattern in the precipitation data for the AP and TN states. Thus, the use of RNN variant models, which are univariate time series forecasting models is suggested for flood prediction since RNN variants capture time dependency in the data and so are better at prediction than other models. Additionally, time-series data is particularly well-suited for RNN [33, 34]. The likelihood of catastrophic flooding increases with heavy precipitation and so precipitation forecasting is necessary to prevent severe casualties.

2.2. DATA PREPROCESSING

Pre-processing is done on the data to remove any null, empty or outlier data or to replace them with the right data before training RNN and ensemble algorithms. Extreme data values that lie outside the observation range are outliers. To eliminate data inconsistencies, outliers and missing numbers are corrected/ filled in and the dataset is formatted for training [35-37]. No similar pre-processing is required because the climatological data obtained NASAPOWER website [14] for AP and TN is free of missing values and outliers.

The precipitation values in the dataset range from 0 to 178.98. To improve model efficiency, min-max scaling as defined in equation (1) is applied to the attribute, precipitation.

$$sc_i = \frac{\min(y_i)}{\max(y_i) - \min(y_i)} \quad (1)$$

MinMaxScalar function in the sklearn package of Python produces the scaled data, SC_i in the range, [0,1] for every i^{th} precipitation observation, y_i. The scaled dataset is divided into training and validation datasets in an 80:20 ratio [36, 37] with the training dataset being used to train the aforementioned RNN variants and ensemble techniques. The performance of the models is evaluated using the validation dataset [35]. The preprocessed univariate time-series dataset is then subjected to the training phase of the RNN algorithms in the Model Building stage.

2.3. MODEL BUILDING

This study proposes to design and analyze precipitation forecast models using time series precipitation data of TN and AP states with variant RNN architectures: the BRNN, LSTM and GRU and ensemble techniques, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU and GRU-SARIMA.

2.3.1. RNN ARCHITECTURES

Recurrent Neural Network is a kind of DL Neural Network; numerous applications with time series data have successfully used the RNN [38]. An RNN uses multiple layers of neurons to construct a model based on training data to forecast unknown or future data. Three basic RNN architectures exist: BRNN [36], LSTM and GRU [39, 40]. A review of these architectures is made in the following sub-sections.

2.3.2. BRNN

The Bidirectional Recurrent Neural Network (BRNN) is a variant of RNN architecture. While unidirectional RNNs can only utilize previous inputs to predict precipitation, BRNN works to increase the forecast accuracy by focusing on the previous and subsequent situations as shown in Fig. 3. It has two RNNs that are oriented in opposite directions and linked to the same output layer. The BRNN receives not only the hidden layer output of the previous moment as an input but also the hidden layer output of the following moment [23].

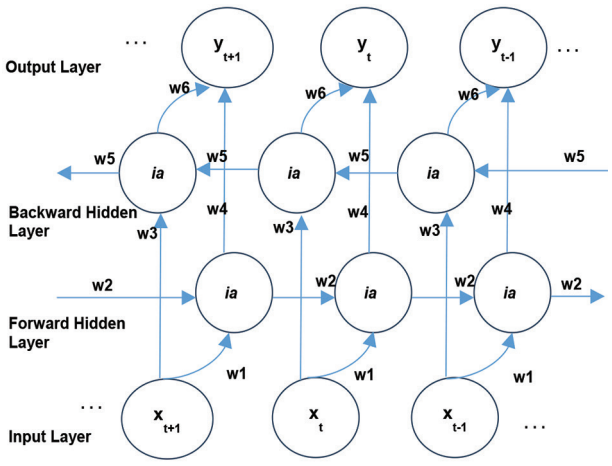


Fig. 3. Structure of BRNN

The values in the hidden and output layer neurons are determined in the forward pass of training based on the equations from (2) to (4).

$$h_t(F) = ia(x_t * w_1(F) + h_{t-1}(F) * w_2(F) + b_h(F) \quad (2)$$

$$h_t(B) = ia(x_t * w_3(B) + h_{t+1}(B) * w_5(B) + b_h(B) \quad (3)$$

$$y_t = h_t(F) * w_4 + h_t(B) * w_6 + b_y \quad (4)$$

Here, x_t is the input with values x_1, x_2, x_3… at time slots, t=1,2,3,…, h_t is the hidden state memory values, h_1, h_2, h_3, ….. at time slots, t=1,2,3,…, ia is the hidden layer activation function, w_1, w_2 are the weights associated with forward hidden layer and w_3, w_4 are the weights associated with the backward hidden layer. $b_h(F)$ and $b_h(B)$ are biases to the forward and backward layers respectively. $h_t(F)$ is the output from the forward hidden layer of $h_t(B)$ is the output from the backward hidden layer. y_t is the precipitation output produced for the instant, 't'. The hyperparameters and optimizers used in the forward and backward passes of the training phase of the BRNN obtained after fine-tuning are mentioned in Table 6 and a discussion on the precipitation forecast is provided in Subsection 3.

2.3.3. LSTM

Sepp Hochreiter and Juergen Schmidhuber introduced the Long Short-Term Memory (LSTM) as an alternative architecture of RNN in 1997 [39, 41]. A typical LSTM structure is made up of a cell unit and three major gates: an input gate, a forgetting gate and an output gate as shown in Fig. 4 (a). The cell unit is a memory unit that can store information for a long period. The memory unit's writing, reading and saving are controlled in order by the input gate, output gate, and forgetting gate. When the forgetting gate outputs 1, the cell unit writes and saves the information; when the forgetting gate outputs 0, the cell unit deletes the saved information; when the input gate outputs 1, the rest of the LSTM reads the cell unit information; and when the input gate outputs 0, the rest of the LSTM writes the contents to the cell unit.

The values of the gates in the hidden layer and output values are determined in the forward pass of training based on the equations from (5) to (11).

$$f_t = \sigma(x_t * u_f + h_{tlstm-1} * w_f) \quad (5)$$

$$\bar{C}_t = ia(x_t * u_c + h_{tlstm-1} * w_c) \quad (6)$$

$$i_t = \sigma(x_t * u_i + h_{tlstm-1} * w_i) \quad (7)$$

$$o_t = \sigma(x_t * u_o + h_{tlstm-1} * w_o) \quad (8)$$

$$C_t = (f_t * C_{t-1} + i_t * \bar{C}_t) \quad (9)$$

$$h_{tlstm} = o_t * ia(C_t) \quad (10)$$

$$y_t = oa(w_t * h_{tlstm} + b) \quad (11)$$

If x_t is the input with values x_1, x_2, x_3… at time slots, t=1,2,3,…, the forget gate, f_t, input gate, it and output

gate, ot values are determined from the equations (5), (7) and (8) respectively. The long-term memory state, C_t and the hidden state memory values, h_{1lstm}, h_{2lstm}, h_{3lstm},…..at time slots, t=1,2,3,…, are determined as in equations (9) and (10) respectively. Here w_f, w_c, w_i and wo are weights associated with the link carrying hidden state information, h_{1lstm} to the forget gate, long-term memory state, input gate and output gate respectively. Also, u_f, u_c, u_i and u_o are the weights associated with links carrying input, x_t to the forget gate, long-term memory state, input gate and output gate respectively. b is the bias associated with output neuron y_t corresponding to the timeslot t. The precipitation forecast of LSTM at different time slots, 't' is determined as in equation (11). σ ia and oa are the *sigmoid*, *tanh* and linear activation functions.

Fig. 4 (b) shows a deep LSTM architecture where the h_{tlstm} value produced by a hidden neuron at a particular level is the input to the next hidden neuron of that level. The output neuron of that level produces the output precipitation value, y_t as a linear function of htlstm value from its immediately previous hidden neuron as in equation (11) using the weight, 'w_t'. The model is trained using the data from NASAPOWER training datasets of AP and TN with the hyperparameters listed in Table 6 for precipitation forecast. A discussion on the test results is provided in Subsection 3.

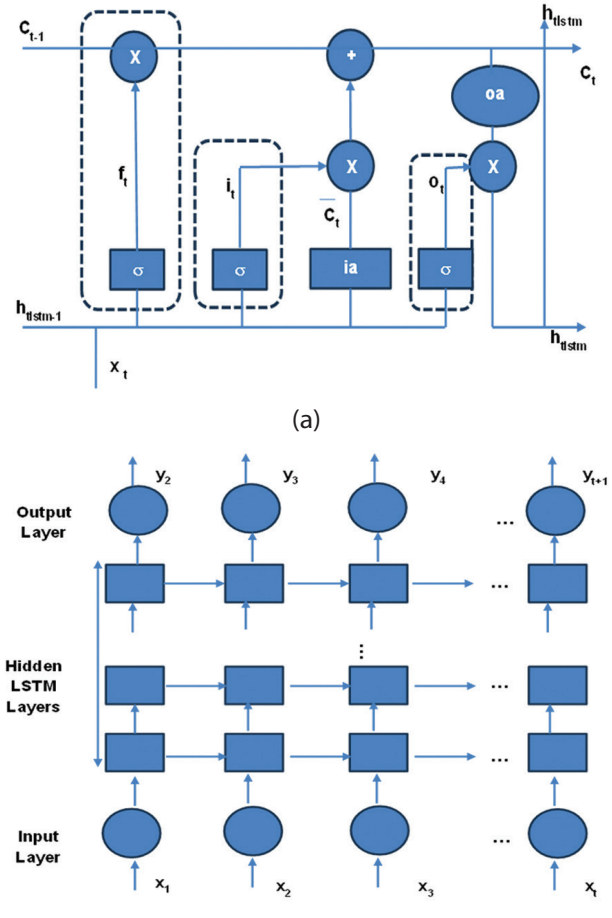


Fig. 4. Structure of a) basic LSTM cell b) deep LSTM

2.3.4. GRU

Another RNN variant is the Gated Recurrent Unit (GRU). It is an upgraded and enhanced version of LSTM which debuted in 2014 [15, 42]. Compared to the three gates of an LSTM, it requires fewer parameters because of the reset gate, rt and update gate, zt. The decision of which information should be transmitted to the output is made using zt and rt, which are vectors as shown in Fig. 5. This study uses the GRU architecture also to predict daily precipitation.

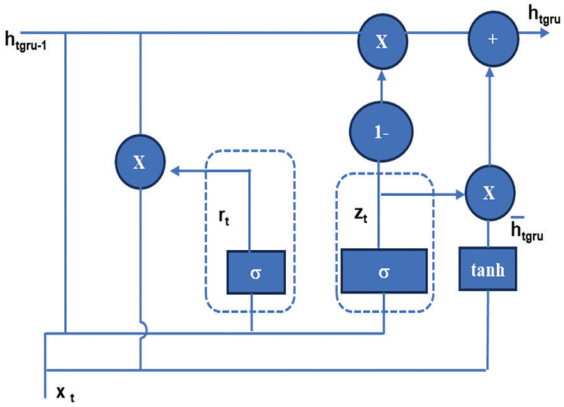


Fig. 5. Structure of basic GRU cell

The values of the gates in the hidden layer and output values are determined in the forward pass of training based on the equations from (12) to (16).

$$z_t = \sigma\left(w_z * \left(h_{tgru-1}, x_t\right)\right) \quad (12)$$

$$r_t = \sigma\left(w_r * \left(h_{tgru-1}, x_t\right)\right) \quad (13)$$

$$\bar{h}_{tgru} = ia\left(w * \left(r_t * \left(h_{tgru-1}, x_t\right)\right)\right) \quad (14)$$

$$h_{tgru} = (1-z_t) * \left(h_{tgru-1} + z_t * \bar{h}_{tgru}\right) \quad (15)$$

$$y_t = oa\left(w_t * h_{tgru} + b\right) \quad (16)$$

If x_t corresponds to the input values x_1, x_2, x_3… at time slots, t=1,2,3,…. Reset gate, r_t and update gate, z_t values are determined from the equations (12) and (13) respectively. The hidden state memory values, h_{1gru}, h_{2gru}, h_{rgru}, ..., at time slots, t=1,2,3,…, are determined as in equations (15). The precipitation forecast of GRU at different time slots is in equation (16). The weights w_z, w_r and w are associated with update gate, reset gate and previous hidden state respectively. b is the bias associated with output neuron, y_t of timeslot t. σ ia and oa are the sigmoid, tanh and linear activation functions.

Deep GRU architecture is similar to deep LSTM as shown in Fig 4 (b), where the h_{tgru} value produced by a hidden neuron at a particular layer is the input to the next hidden neuron of that level. The output neuron of that level produces the output precipitation value, y_t as a linear function of h_{tgru} value from its immediately previous hidden neuron as in equation (16). In eqn. (16), wt is the weight associated with the output neuron, y_t.

The model is trained using the training datasets from AP and TN with the hyperparameters listed in Table 6 which are obtained by analyzing various hyperparameters and optimizers. Also, a discussion of the predicted precipitation is provided in Subsection 3.

2.3.5. ENSEMBLE ARCHITECTURE

This section discusses the ensemble models that incorporate the RNN variant models: BRNN, LSTM and GRU. Averaging, Bagging, and Stacking are the three ensemble learning techniques [43]. Ensemble learning techniques are often trained using many models on the same dataset utilizing each learned model to generate a forecast and combining the results of the individual model to produce the final result or forecast [43]. In this work, an average ensemble of selected RNN models is determined based on the results of the models to produce accurate precipitation forecasts as shown in Fig. 6.

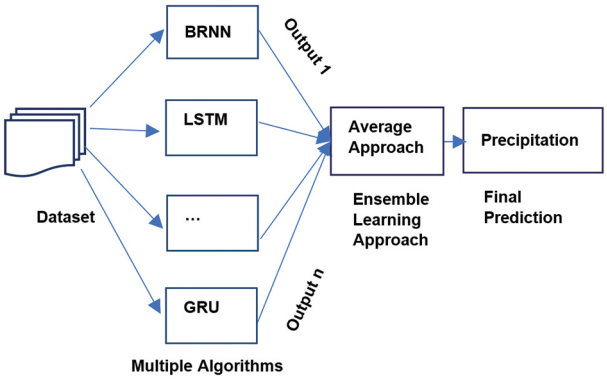


Fig. 6. General Workflow for Ensemble Techniques

The different ensemble models used in the analysis are BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU as shown in Fig. 7. The optimized hyperparameters used by LSTM, GRU and BRNN as mentioned in Table 6 are used by the base models of the ensemble techniques also. The performance of the models BRNN, LSTM, GRU, BRNN-GRU, BRNN- LSTM, LSTM-GRU and BRNN-LSTM-GRU are provided in Table 7 and 8.

The RNN variant with the lowest error, GRU is also used to create a hybrid ensemble model with statistical SARIMA for precipitation forecasting as shown in Fig. 8. SARIMA is a Seasonal AutoRegressive Integrated Moving Average model for time-series data-based forecasting that uses past values. SARIMA is an advanced version of AutoRegressive Integrated Moving Average (ARIMA) [7] for Time Series forecasting based on its past values, lags and lagged forecast error values along with the seasonal characteristics of the time-series data with seasonal patterns.

The SARIMA model has the parameters: the order of the 'Auto Regressive' (AR) phrase, the number of lags to be utilized as predictors and the order of the moving average along with the parameters of seasonal characteristics which includes: seasonal autoregressive order, seasonal difference order, seasonal moving average order and the periodicity of seasons [10]. The precipitation is found by averaging the forecasts of the GRU model and the SARIMA model. The ensemble model is trained using historical precipitation data of TN and AP from 2002 to 2022. By leveraging its learned patterns, it predicts the precipitation for the next 30 days, providing valuable insights for planning and decision-making against floods. Additionally, the performance of this hybrid ensemble model is assessed in terms of MAE, MSE, RMSE and RMSLE. In Section 3, the outcomes of these RNN variant models and ensemble models are examined in terms of evaluation metrics, MAE, MSE, RMSE, and RMSLE. The values are shown in Tables 7 and 8.

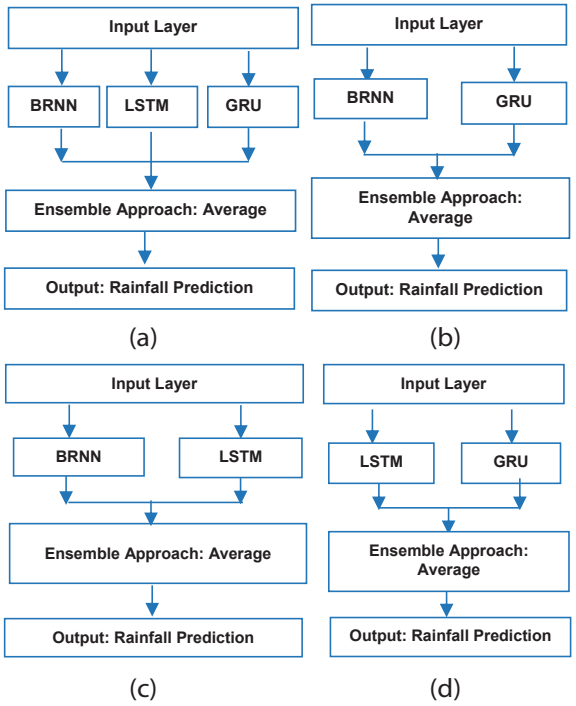


Fig. 7. Architecture of RNN-based Ensemble models a) BRNN-LSTM-GRU b) BRNN-GRU c) BRNN-LSTM d) LSTM-GRU

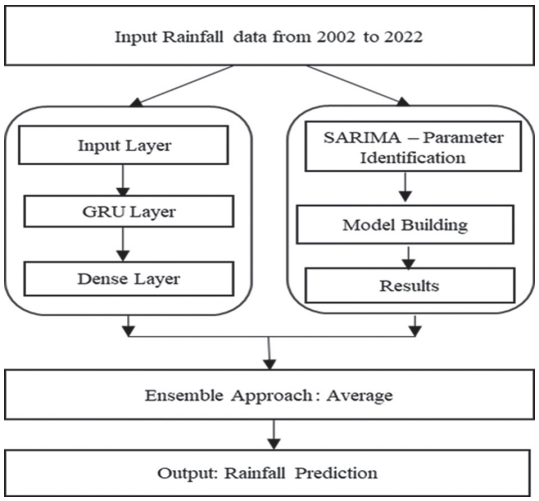
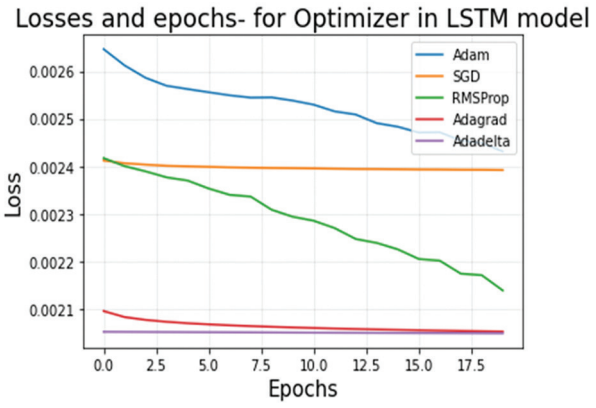
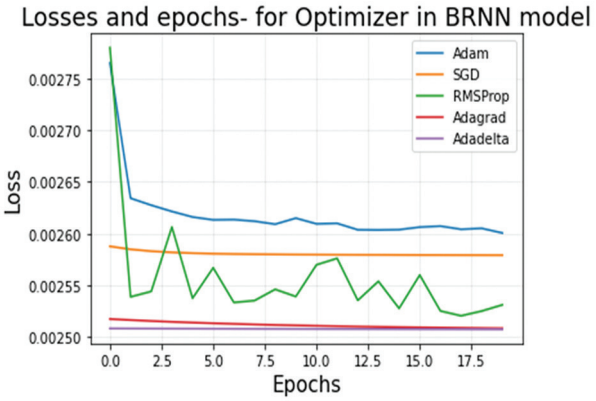


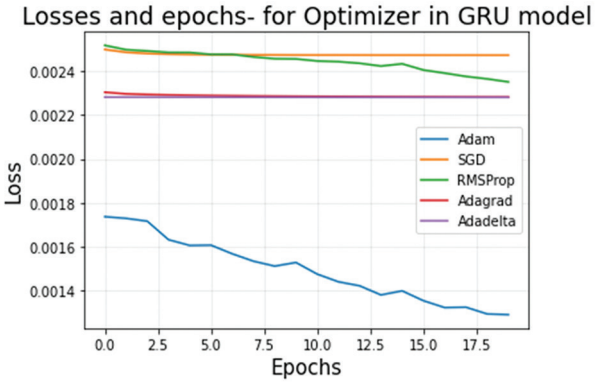
Fig. 8. Block diagram showing GRU-SARIMA ensemble model



(a)



(b)



(c)

Fig. 9. Optimizer based losses in RNN variants a) LSTM b) BRNN and c) GRU

These models are developed using the DL and essential packages in Python namely TensorFlow, NumPy, Pandas, Matplotlib, Sklearn and Keras. Appropriate hyperparameters are essential to achieve the best results from the RNN variants and the ensemble models [43-45]. Hyperparameters are variables that control the neural network architecture and performance during training. Some of the hyperparameters are the number of layers, activation functions, loss function, batch size and optimizer. Optimizers and hyperparameters have been studied in this work to improve the accuracy of the precipitation forecasts from the RNN and ensemble models. The working steps for selecting hyperparameters and optimizers are explained in the following paragraph.

The RNNs variants used in this study are compared against various optimization models including Adaptive Moment Estimation (Adam) [46], Stochastic Gradient Descent (SGD) [46], Adaptive Gradient Descent (AdaGrad) [46], Extension of AdaGrad (Adadelta) [40, 46] and Root Mean Square Propagation (RMSProp) [40, 46] to determine which optimization model offers the best learning and prediction. For the same set of hyperparameters as listed in Table 2, different optimizers are tried for all the RNN variants. The performance is assessed in terms of RMSLE by making forecasts on the validating dataset as shown in Table 3. The losses are illustrated in Fig. 9 a) LSTM, b) BRNN, and c) GRU respectively for different epochs of the training.

Table 2. List of Hyperparameters for Selecting Optimizer

Hyperparameters	Value of Hyperparameters
Batch Size	32
No of epochs	20
No of Hidden Layers	1
Hidden Units	100
Activation Function	Tanh
Output-Units	30
Output-Layer-Activation-Function	Linear
Loss Function	MAE

Of all the optimizers, the Adam optimizer generated smaller and steadily decreasing losses while using the training dataset. Unlike Adam, the other optimizers delivered constant losses and substantial forecast errors. Even while the RMSProp offered continuously decreasing losses, the losses in the BRNN model fluctuate. As a result, in this study, the Adam optimizer is employed to forecast precipitation from all the RNN variants and ensemble models. Also, to select the best set of hyperparameters for RNN variants such as BRNN, LSTM, and GRU, a list of experiments with different hyperparameter combinations as mentioned in Table 4 are performed on all RNN variants.

Table 3. Evaluation of Optimizers based on RMSLE

Model	Optimizers				
	Adam	SGD	RMSProp	AdaGrad	Adadelta
LSTM	0.0017	0.0017	0.0020	0.0019	0.0019
GRU	0.0018	0.0018	0.0018	0.0018	0.0018
BRNN	0.0018	0.0018	0.0018	0.0018	0.0018

The RNN variants generated results as shown in Table 5 for different trials. The RMLSE values for BRNN, LSTM, and GRU in the trials from 1 to 4 are 0.26, 1.42, 0.26; 0.58, 1.39, 0.30; 0.57, 2.01, 0.48 and 0.97,1.10, 0.95 respectively. All trials from 1 through 4 are compared to select the best hyperparameters. The trials 1 and 2 are compared to select the best batch; trials 3 and 4 are compared to select the best epoch and the trials 1 and

4 are compared to select the best number of hidden layers. The performances of the different RNN variants in the above experiments are compared as shown in Fig.10 a) to d) of Section 3 in terms of MAE, MSE, RMSE and RMSLE. The most effective optimizer and hyperparameters as identified from the experiments and listed in Table 6 are selected for further training and testing to predict precipitation.

Table 4. List of Hyperparameters and Optimizer values for selecting best hyperparameters

Hyperparameters	Trial 1	Trial 2	Trial 3	Trial 4
Number of Hidden Layers	3	3	1	1
Hidden units	128,256, 128	128, 256,128	128	128
Number of epochs	100	100	300	100
Batch size	32	64	32	32
Activation function	Tanh	Tanh	Tanh	Tanh
Optimizer	Adam	Adam	Adam	Adam
Loss function	MSE	MSE	MSE	MSE
Output-Units	30	30	30	30
Output-Layer-Activation-Function	Linear	Linear	Linear	Linear

Table 5. Comparison on the Performance of RNN variants in different trials

Model	Experiment	MAE	MSE	RMSE	RMSLE
BRNN		0.0039	0.0621	0.0350	0.0028
LSTM	Trial1	0.0003	0.0186	0.0107	0.0002
GRU		0.0001	0.0123	0.0072	0.0001
BRNN		0.0039	0.0622	0.0337	0.0029
LSTM	Trial2	0.0005	0.0241	0.0134	0.0004
GRU		0.0004	0.0216	0.0124	0.0003
BRNN		0.0038	0.0615	0.0348	0.0027
LSTM	Trial 3	0.0001	0.0122	0.0068	0.0001
GRU		0.0003	0.0193	0.0113	0.0002
BRNN		0.0031	0.0559	0.0269	0.0023
LSTM	Trial 4	0.0018	0.0426	0.0223	0.0014
GRU		0.0014	0.0377	0.0206	0.0011

Table 6. List of Hyperparameter values and optimizer selected for building BRNN, LSTM and GRU models

Hyperparameters	Value
Number of Hidden Layers	3 (128,256,128 units)
Number of epochs	300
Batch size	32
Activation function	Tanh
Optimizer	Adam
Loss function	MSE
Output-Units	30
Output-Layer-Activation-Function	Linear

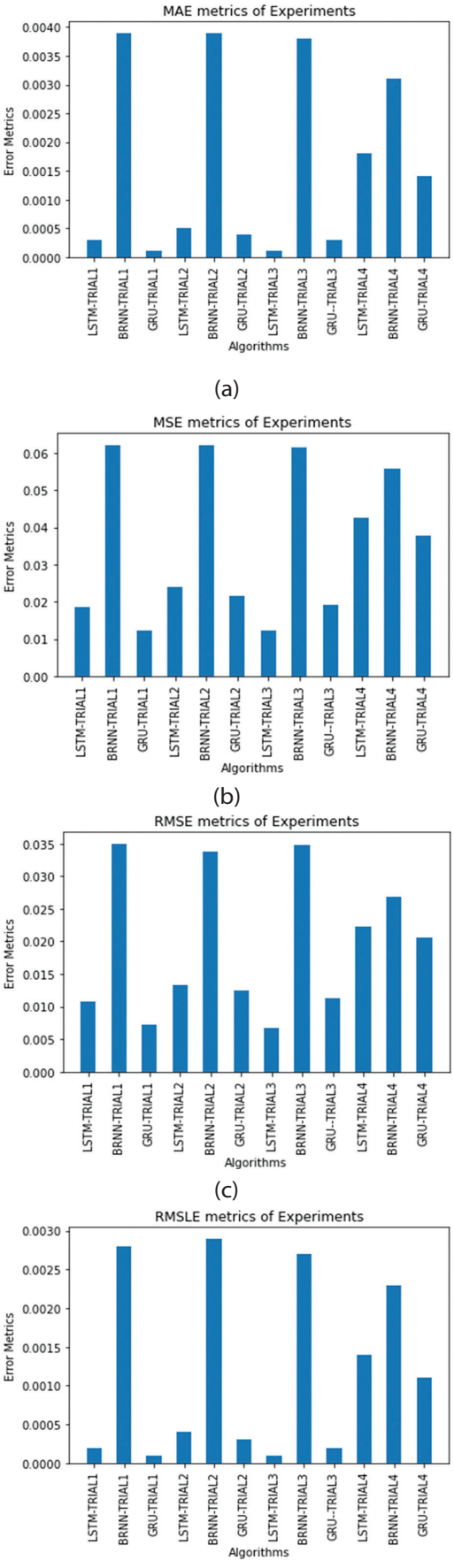


Fig. 10. Bar plot showing the performances of RNN variants for different trials in terms of a) MAE b) MSE c) RMSE d) RMSLE

3. EXPERIMENTAL RESULTS AND DISCUSSION

This study underscores the importance of precipitation forecast to mitigate the negative effects of flood damage. Time-series data are often used to make predictions or forecasts of future values based on historical observations. The time-series data is obtained from the website, https://power.larc.nasa.gov/ [14] to forecast precipitation. Fig. 2 illustrates time-series data for days, weeks and months for AP and TN. However, day-wise data is used to train the deep learning RNN algorithms and ensemble techniques to predict the precipitation for the next 30 days. The model is created from the past 60 days' precipitation data to predict the precipitation of the next 30 days.

Experiments are conducted on an Intel Core i7 processor running at 2.70 GHz with 16GB of RAM using numerical and DL libraries of Python. TensorFlow, NumPy, Pandas, Matplotlib, Keras, and Sklearn are some of the packages used from Python. RNN based precipitation forecasting models are evaluated using the performance assessment metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Root Mean Squared Logarithmic Error (RMSLE) values [30, 47]. The Mean Absolute Error (MAE) is the average of the absolute error difference as defined in Equation (17).

$$MAE = \frac{\sum_{i=1}^{m}|y_{ai} - y_{pi}|}{m} \quad (17)$$

The square root of the Mean Square Error (MSE) is referred to as the RMSE and is defined by Equation (18).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{ai} - y_{pi})^2} \quad (18)$$

where

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_{ai} - y_{pi})^2 \quad (19)$$

An RMSE variation that computes the logarithmic difference between the predicted and actual values is RMLSE. It is defined in equation (20).

$$RMSLE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\log(y_{ai} + 1) - \log(y_{pi} + 1))^2} \quad (20)$$

In all the above equations, the predicted value and the actual value corresponding to the i^{th} observation are denoted as y_{pi} and y_{ai} respectively. m is the number of records in the test dataset.

3.1. RESULTS AND DISCUSSION

In this work, for precipitation forecast, time-series based climatological data from 2002 to 2022 are downloaded from the NASAPOWER website. Subsequently, the data preprocessing operations are carried out. Afterward, the data is split in the ratio, 80:20 for data training and validation. The most effective hyperparameters and optimizer as identified and mentioned in Table 6 are chosen in building the RNN and ensemble models: BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU.

The test accuracy of predictions made by the models are evaluated using MSE, RMSE, MAE, and RMSLE. These evaluation metric values are presented in Table 8 and 9 and in Fig. 11 a) to d) for AP and TN respectively. Finally, using the RNN model, GRU which produces lowest error, a hybrid ensemble model is created with the statistical model, SARIMA to predict precipitation. The accuracy of this model is listed in Table 7 and 8 for AP and TN respectively. The GRU model produces improved performance than all the other models under comparison with a net RMSLE value of 0.255 and 0.152 for AP and TN datasets respectively. The ensemble model, LSTM-GRU proves to be the next best model from among all the ensemble models under comparison with a net RMSLE value of 0.364 and 0.379 for AP and TN datasets respectively. The best performing RNN model, GRU when ensembled with the existing statistical model SARIMA produces an RMSLE value of 0.754 and 1.677 respectively for AP and TN. The results of the proposed GRU model is also compared in terms of error metrics with those from publications as shown in Table 9. From the analysis, of all the models, the best performing GRU model appears reliable for flood defense due to heavy precipitation. The comparative analysis demonstrates that the GRU as identified from the proposed methodology outperforms all other models and techniques in terms of RMSE values.

Table 7. Comparison on the Performance of different RNN models and ensemble techniques on Test Dataset from AP

Models	MAE	MSE	RMSE	RMSLE
LSTM	1.041	2.113	1.453	0.555
BRNN	4.255	30.922	5.561	2.448
GRU	0.635	0.558	0.747	0.255
BRNN-LSTM	2.351	10.176	3.190	1.383
BRNN-GRU	2.240	8.843	2.974	1.305
LSTM-GRU	0.766	0.933	0.966	0.364
BRNN-LSTM-GRU	2.972	15.683	1.453	1.740
GRU-SARIMA	1.461	1.278	1.528	0.754

Table 8. Comparison on the Performance of different RNN models and ensemble techniques on Test Dataset from TN

Models	MAE	MSE	RMSE	RMSLE
LSTM	1.482	6.221	2.494	0.663
BRNN	3.296	16.772	2.494	1.735
GRU	0.519	0.352	0.593	0.152
BRNN-LSTM	2.160	6.945	0.593	1.118
BRNN-GRU	1.801	4.843	2.201	0.889
LSTM-GRU	0.910	2.130	1.460	0.379
BRNN-LSTM-GRU	2.525	9.209	2.494	1.328
GRU-SARIMA	2.905	8.905	2.484	1.677

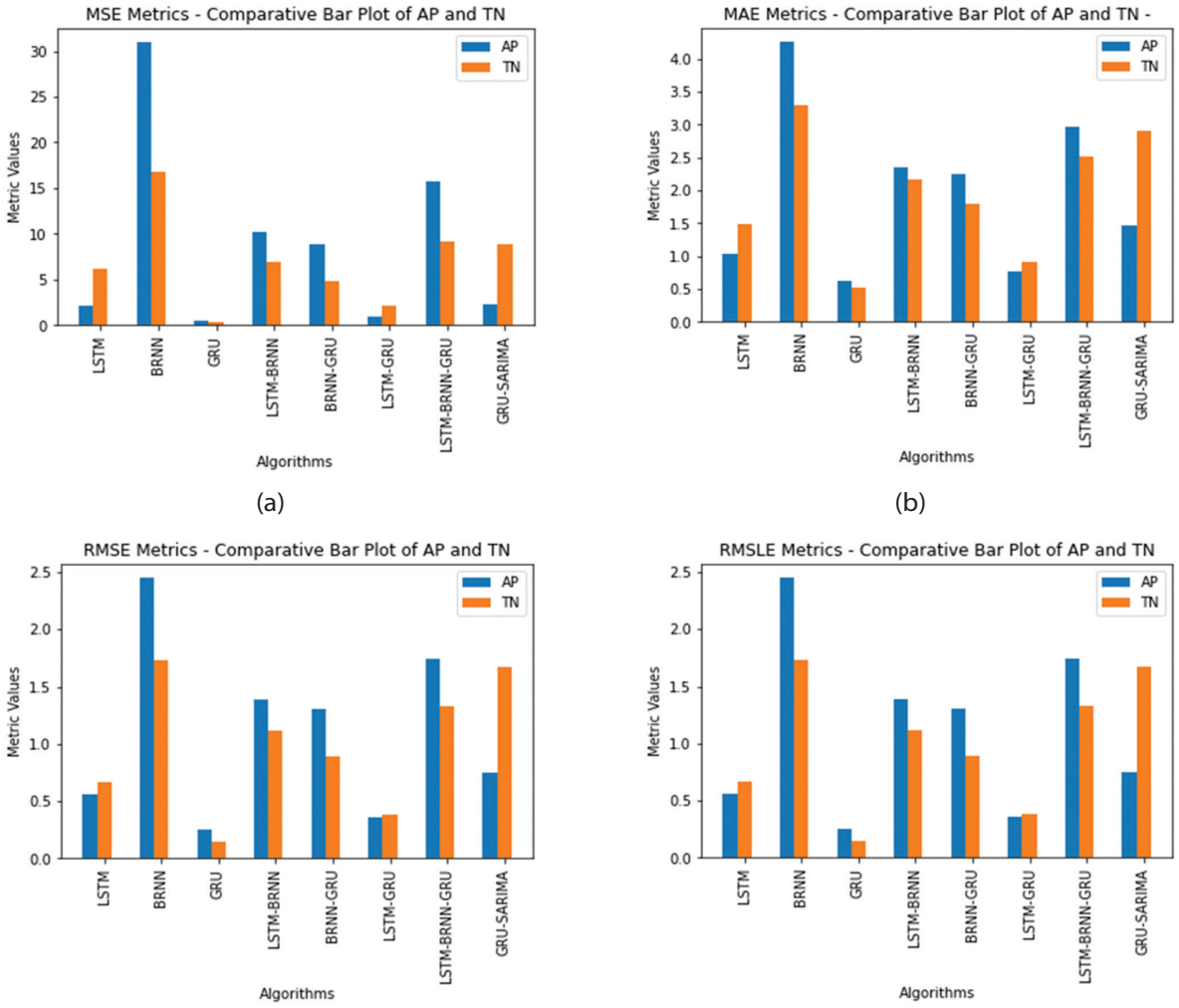


Fig. 11. Bar plot showing the performances of different of RNN variants and ensemble techniques on Test Dataset from TN and AP in terms of a) MAE b) MSE c) RMSE d) RMSLE

Table 9. Comparison of results from different algorithms

Authors	Region & Dataset	Algorithms	Best RMSE Value
S Aswin, et al. (2018) [26]	Geographic location 10368 - [26]	LSTM, CNN	LSTM- 2.55, CNN-2.44
Y.O. Ouma, et al. (2020) [29]	Kenya - [29]	LSTM, WNN	LSTM -17.22 WNN- 14.64
Samad, A. et al. (2020) [30]	Australia - [30]	LSTM	LSTM - 5.30
E Gbenga Dada et al. (2021) [21]	India - [21]	FFNN, RNN, ENN, CFNN	ENN -6.36
Proposed Methodology-GRU	India – [14]	RNN variant and Ensemble	GRU - 0.593

4. CONCLUSION

Predicting precipitation is an effective flood defense method that helps minimize the impact of high precipitation events and safeguard vulnerable areas. In this study, RNN-based precipitation forecast algorithms were trained and tested using the time-series-based climatological data of heavy rainfall receiving South Indian states of Tamil Nadu, Andhra Pradesh. The DL algorithms and the ensemble techniques: BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU and GRU-SARIMA were trained with the best set of hyperparameters and optimizers identified experimentally. In addition, the performances of these models were assessed in terms of MAE, MSE, RMSE, and RMSLE values. The GRU model proved to be the most effective model among all the models with net RMSLE values of 0.225 and 0.152 for AP and TN datasets respectively. The ensemble model, LSTM-GRU proved the next best model from among all the models under comparison with net RMSLE values of 0.364 and 0.379 for AP and TN datasets respectively. Also, hybrid ensemble model GRU-SARIMA proved to be the effective model, with net RMSLE values of 0.754 and 1.677 for AP and TN datasets respectively. Thus, the analysis concludes GRU model for precipitation predictions from time-series based climatological data as a mechanism to precaution flood. When the precipitation forecast exceeds the threshold of 64.5 mm/day as mentioned in [8-10], it is a flood alarm for planning the disaster. This work can be

extended to prediction models that combine multivariate datasets, data from multiple sources for improved precipitation forecasts.

Acknowledgment

The data was downloaded from the website https://power.larc.nasa.gov [14]. The website associates with the NASA Earth Science/Applied Science Programme and NASA Langley Research Centre (LaRC) POWER Project.

5. REFERENCES:

[1] S. N. Jonkman, "Global perspectives on loss of human life caused by floods", Natural Hazards, Vol. 34, No. 2, 2005, pp. 151-175.

[2] O. Singh, M. Kumar, "Flood occurrences, damages, and management challenges in India: a geographical perspective", Arabian Journal of Geosciences, Vol. 10, No. 102, 2017, pp. 1-19.

[3] I. Yucel, A. Onen, K.K. Yilmaz, D.J. Gochis, "Calibration and evaluation of a flood forecasting system: Utility of numerical weather prediction model, data assimilation and satellite-based rainfall", Journal of Hydrology, Vol. 523, 2015, pp. 49-66.

[4] O. Singh, M. Kumar, "Flood events, fatalities and damages in India from 1978 to 2006", Natural hazards, Vol. 69, 2013, pp. 1815-1834.

[5] Central Water Commission, Ministry of Jal shakti, Department of Water Resources, River Development and Ganga Rejuvenation, https://cwc.gov.in/flood-damage-statistics-statewise-and-country-whole-during-1953-2020 (accessed: 2023)

[6] Government of India - India Meteorological Department, https://mausam.imd.gov.in/Forecast /marquee_data/Statement_climate_of_india_2022_final.pdf (accessed: 2023)

[7] A. S. Abdullah, B. N. Ruchjana, I. G. N. M. Jaya, Soemartini, "Comparison of SARIMA and SVM model for rainfall forecasting in Bogor city", Journal of Physics: Conference Series of ICW-HDDA-X 2020, Vol. 1722, No. 1, 2021, pp. 1-8.

[8] P. Guhathakurta, O. P. Sreejith, P. A. Menon, "Impact of climate change on extreme rainfall events and flood risk in India", Journal of Earth System Science, Vol. 120, No. 3, 2011, pp. 359-373.

[9] I. G. Tunas, H. Azikin, G. M. Oka, "Impact of Extreme Rainfall on Flood Hydrographs", Proceedings of the 2nd International Conference on Hazard Mitigation in Geographic and Education Perspectives, Indonesia, 11-12 September 2020, pp. 1-6.

[10] M. Rajeevan, J. Bhate, A. K. Jaswal, "Analysis of variability and trends of extreme rainfall events over India using 104 years of gridded daily rainfall data", Geophysical Research Letters, Vol. 35, No. 18, 2008, pp.1-6.

[11] L. J. Bracken, N. J. Cox, J. Shannon, "The relationship between rainfall inputs and flood generation in south–east Spain", Hydrological Processes: An International Journal, Vol. 22, No. 5, 2008, pp. 683-696.

[12] Wikipedia, "South India", https://en.wikipedia.org/wiki/South_India (accessed: 2023)

[13] J. Subha, S. Saudia, "An Exploratory Data Analysis on SDMR Dataset to Identify Flood-Prone Months in the Regional Meteorological Subdivisions", Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022, India, 6-7 July 2022, pp. 595-617.

[14] NASA, "Prediction of Worldwide Energy Resources", https://power.larc.nasa.gov/ (accessed: 2023)

[15] J. T. de Castro, G. M. Salistre Jr, Y. C. Byun, B. D. Gerardo, "Flashflood prediction model based on multiple regression analysis for decision support system", Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 23-25 October 2013, pp. 23-28.

[16] E. Jumin, F. B. Basaruddin, Y. B. M. Yusoff, S. D. Latif, A. N. Ahmed, "Solar radiation prediction using boosted decision tree regression model: A case study in Malaysia", Environmental Science and Pollution Research, Vol. 28, 2021, pp. 26571-26583.

[17] J. Subha, S. Saudia, "Integrating Regression Models and Climatological Data for Improved Precipitation Forecast in Southern India", International Journal of Advanced Computer Science and Applications, Vol. 14, No. 5, 2023, pp. 626-638.

[18] A. U. Azmi, A. F. Hadi, D. Anggraeni, A. Riski, "Naive bayes methods for precipitation prediction classification in Banyuwangi", Journal of Physics: Conference Series, Vol. 1872, No. 1, 2021, pp. 1-8.

[19] Y. Dash, S. K. Mishra, B. K. Panigrahi, "Rainfall prediction for the Kerala state of India using artificial

intelligence approaches", Computers and Electrical Engineering, Vol. 70, 2018, pp. 66-73.

[20] M. Rezaeianzadeh, H. Tabari, A. Arabi Yazdi, S. Isik, L Kalin, "Flood flow forecasting using ANN, ANFIS and regression models", Neural Computing and Applications, Vol. 25, No. 1, 2014, pp. 25-37.

[21] E. G. Dada, H. J. Yakubu, D. O. Oyewola, "Artificial neural network models for precipitation prediction", European Journal of Electrical Engineering and Computer Science, Vol. 5, No. 2, 2021, pp. 30-35.

[22] R. Dey, F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks", Proceedings of the IEEE 60th international Midwest symposium on circuits and systems, Boston, MA, USA, 6-9 August 2017, pp. 1597- 1600.

[23] H. Song, H. Choi, "Forecasting Stock Market Indices Using the Recurrent Neural Network Based Hybrid Models: CNN-LSTM, GRU-CNN, and Ensemble Models", Applied Sciences, Vol. 13, No. 7, 2023, pp. 1-26.

[24] Y. Ji, B. Gong, M. Langguth, A. Mozaffari, X. Zhi, "CLGAN: a generative adversarial network (GAN)-based video prediction model for precipitation nowcasting", Geoscientific Model Development, Vol. 16, No. 10, 2023, pp. 2737-2752.

[25] R. Venkatesh, C. Balasubramanian, M. Kaliappan, "Precipitation prediction using generative adversarial networks with convolution neural network", Soft Computing, Vol. 25, 2021, pp. 4725-4738.

[26] S. Aswin, P. Geetha, R. Vinayakumar, "Deep learning models for the prediction of precipitation", Proceedings of the International Conference on Communication and Signal Processing, Chennai, India, 3-5 April 2018, pp. 657-661.

[27] D. Z. Haq, D. C. R. Novitasari, A. Hamid, N. Ulinnuha, Arnita, Y. Farida, R. R. D. Nugraheni, R. Nariswari, Illham, H. Rohayani, R. Pramulya, A. Widjayanto, "Long short-term memory algorithm for precipitation prediction based on El-Nino and IOD data", Proceedings of 5th International Conference on Computer Science and Computational Intelligence 2020, Indonesia, 19-20 November 2020, pp. 829-837.

[28] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, H. Xinhua, "Prediction of short-time precipitation based on deep learning", Mathematical Problems in Engineering, Vol. 2021, 2021, pp. 1-8.

[29] Y. O. Ouma, R. Cheruyot, A. N. Wachera, "Precipitation and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: case study of Nzoia hydrologic basin", Complex & Intelligent Systems, Vol. 8, 2022, pp. 213-236.

[30] A. Samad, Bhagyanidhi, V. Gautam, P. Jain, Sangeetha, K. Sarkar, "An approach for precipitation prediction using long short term memory neural network", Proceedings of the IEEE 5th International Conference on Computing Communication and Automation, Greater Noida, India, 30-31 October 2020, pp. 190-195.

[31] M. Saha, P. Mitra, R. S. Nanjundiah, "Deep learning for predicting the monsoon over the homogeneous regions of India", Journal of Earth System Science, Vol. 126, No. 54, 2017, pp. 1-18.

[32] W. Suparta, A. A. Samah, "Rainfall prediction by using ANFIS times series technique in South Tangerang, Indonesia", Geodesy and Geodynamics, Vol. 11, No. 6, 2020, pp. 411-417.

[33] L. Zhang, R. Wang, Z. Li, J. Li, Y. Ge, S. Wa, S. Huang, C Lv, "Time-Series Neural Network: A High-Accuracy Time-Series Forecasting Method Based on Kernel Filter and Time Attention", Information, Vol. 14, No. 9, 2023, p. 500.

[34] I. V. Necesito, D. Kim, Y. H. Bae, K. Kim, S. Kim, H. S. Kim, "Deep Learning-Based Univariate Prediction of Daily Rainfall: Application to a Flood-Prone, Data-Deficient Country", Atmosphere, Vol. 14, No. 4, 2023, p. 632.

[35] B. Rekabdar, D. L. Albright, J. T. McDaniel, S. Talafha, H. Jeong, "From machine learning to deep learning: A comprehensive study of alcohol and drug use disorder", Healthcare Analytics, Vol. 2, 2022, pp. 1-18.

[36] C. Fan, M. Chen, X. Wang, J. Wang, B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data", Frontiers in Energy Research, Vol. 9, 2021, pp. 1-17.

[37] F. Qayyum, D. H. Kim, S. J. Bong, S. Y. Chi, Y. H. Choi, "A Survey of Datasets, Preprocessing, Modeling Mechanisms, and Simulation Tools Based on AI for Material Analysis and Discovery", Materials, Vol. 15, No. 4, 2022, p. 1428.

[38] P. Li, J. Zhang, P. Krebs, "Prediction of flow based on a CNN-LSTM combined deep learning approach", Water, Vol. 14, No. 993, 2022, pp. 1-13.

[39] H. Bohan, B. Yun, "Traffic flow prediction based on BRNN", Proceedings of the IEEE 9th International Conference on Electronics Information and Emergency Communication, Beijing, China, 12-14 July 2019, pp. 320-323.

[40] A. Dalli, "Impact of hyperparameters on Deep Learning model for customer churn prediction in telecommunication sector", Mathematical Problems in Engineering, Vol. 2022, 2022, pp. 1-11.

[41] V. Linardos, M. Drakaki, P. Tzionas, Y. L. Karnavas, "Machine learning in disaster management: recent developments in methods and applications", Machine Learning and Knowledge Extraction, Vol. 4, No. 2, 2022, pp. 446-473.

[42] J. Siłka, M. Wieczorek, M. Woźniak, "Recurrent neural network model for high- speed train vibration prediction from time series", Neural Computing and Applications, Vol. 34, No. 16, 2022, pp. 13305-13318.

[43] R. Odegua, "An empirical study of ensemble techniques (bagging, boosting and stacking)", Proceedings of the Conference Deep Learn, 2019, pp. 1-10.

[44] Z. H. Kilimci, "Ensemble Regression-Based Gold Price (XAU/USD) Prediction", Journal of Emerging Computer Technologies, Vol. 2, No. 1, 2022, pp. 7-12.

[45] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, A. Krishnan, "Flood prediction based on weather parameters using deep learning", Journal of Water and Climate Change, Vol. 11, No. 4, 2020, pp. 1766-1783.

[46] M. N. Halgamuge, E. Daminda, A. Nirmalathas, "Best optimizer selection for predicting bushfire occurrences using deep learning", Natural Hazards, Vol. 103, No. 1, 2020, pp. 845-860.

[47] D. Chicco, M. J. Warrens, G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation", PeerJ Computer Science, Vol. 7, 2021, pp. 1-24.