Feature Selection and Optimization Based Deep Learning for Rainfall Prediction

P. Vijaya[1](✉), Satish Chander[2], Praba Palanisamy[3], Alycia Sebastian[4], and Joseph Mani[4]

[1] Modern College of Business and Science, Bowshar, Oman
Vijaya.Padmanabha@mcbs.edu.om
[2] Birla Institute of Technology, Mesra, Ranchi, India
satishchander@bitmesra.ac.in
[3] University of Technology and Applied Sciences, Muscat, Oman
praba.p@hct.edu.om
[4] Modern College of Business and Science, Bowshar, Oman
{alycia.sebastian,drjosephmani}@mcbs.edu.om

Abstract. Rainfall hugely impacts every aspect of human life, such as transportation, agriculture, water management, and so on. It also is a grave cause of several natural calamities, like landslides, floods, and drought, which pose a serious threat to the well-being of individuals. These concerns have necessitated the need for devising an effective technique to predict rainfall, which enables the undertaking of effective preventive measures. Several works have focused on developing efficient rainfall forecasting techniques; however, the uncertain nature of rainfall and the lack of rainfall data limit their effectiveness. This paper proposes an efficient rainfall prediction strategy using an optimized Deep Learning approach. Here, prediction is carried out using a Deep Long Short Term Memory network based on the time series data of the rainfall. Further, the prediction efficiency is enhanced by the utilization of the Circle Inspired Optimization Algorithm for the weight optimization of the Deep Long Short Term Memory. Experimental results show that the devised Circle Inspired Optimization Algorithm-Deep Long Short Term Memory reveals enhanced performance by attaining a minimal value of Relative Absolute Error at 0.023 Mean Square Error of 0.151, and Root Mean Square Error of 0.389.

Keywords: Rainfall prediction · time-series data · Deep Learning · Technical indicators · optimization

1 Introduction

Rainfall is the most dominant meteorological factor in the various facets of day-to-day life. Rainfall can cause several noteworthy socio-economic effects ranging from interruptions in transport services to the destruction of infrastructure in case of floods. Extreme events, like floods, are the result of changes in climatic conditions, which are prone to happen more commonly, and these events will cause numerous natural

N. Bacanin and H. Shaker (Eds.): ICIITB 2022, ACSR 104, pp. 235–249, 2023.
https://doi.org/10.2991/978-94-6463-110-4_17

calamities in the forthcoming years [1]. Further, the variation in weather can cause a potential surge in air pollution during summer and winter [2]. The increase in air pollution levels can lead to severe health issues, like asthma and other lung diseases [3, 4]. Moreover, it is an essential factor in agriculture, thus impacting the economic condition of any country. In addition, it is a crucial factor impacting various fields, such as hospitality [5, 6], forestry, electricity generation, and so on. Rainfall prediction has to be accomplished in advance to enable effective harvesting of rainwater in the event of water shortage. Further, it is highly crucial in handling natural calamities, like avalanches, mass movements, floods, and landslides. The aftermath of natural disasters occurring due to rainfall has an enduring impact on the culture of any region and hence, effective rainfall prediction can aid in taking preventive actions against natural calamities and alleviating them [7, 8].

Practically, rainfall prediction is regarded as one of the highly difficult tasks and has been carried out globally by multiple researchers. A major aspect restricting the effectiveness of rainfall prediction is mainly the uncertain and chaotic aspect of rainfall [9]. The majority of the hydrological processes normally depict a higher degree of spatial and temporal variance that causes non-stationaries in the observed hydrological information [10]. Accurate rainfall prediction offers an exact approximation of the risks associated and helps in realizing efficient mitigation strategies [11]. The challenges in rainfall forecasting are mainly due to the rainfall's seasonal nature and measure. Normally, rainfall prediction is executed using two strategies: dynamic and empirical schemes. The dynamic techniques utilize statistical as well as physical models to forecast seasonal rainfall, while the empirical schemes utilize the relation among the historical information. Artificial Neural Networks (ANN) and regression models use the concept of the empirical scheme in forecasting [12]. Furthermore, Deep learning (DL) schemes and optimization algorithms, like Firefly Algorithm [13–16], Multi-Swarm Algorithm [17], particle swarm optimization [18], sine cosine algorithm [19], Moth Flame Optimizer [20] have shown promising results for solving complicated issues with less computational complexities and hence used in dealing with hydrological variable prediction at different spatial and temporal scales [21].

This paper devises an effective rainfall prediction technique based on the input time series data of rainfall. Further, the technique utilizes several technical indicators to identify the most relevant information in the input data. From these features, Tanimoto and Dice similarity are employed to determine the prominent features, which are provided to the Deep Long Short Term Memory (Deep LSTM) for prediction. The weight parameters of the Deep LSTM are adapted based on the Circle Inspired Optimization Algorithm (CIOA).

The key contribution of this work is as follows,

- ***Devised CIOA-Deep LSTM for rainfall prediction:*** Here, rainfall prediction is carried out using the Deep LSTM based on the optimal technical indicators acquired from the input time series data. Moreover, the CIOA is utilized to enhance prediction efficiency by optimizing the weight parameters of the Deep LSTM.

The organization of the rest of the work is as follows: Sect. 2 presents the existing rainfall prediction strategies, Sect. 3 details the current work, with the experimental outcomes portrayed in Sect. 4, and the conclusion is elaborated on in Sect. 5.

2 Motivation

Rainfall prediction techniques have been addressed for a long time, and numerous works have been proposed to carry out the task of rainfall prediction. The time-series data utilized in the forecasting process makes it highly difficult in estimating the rainfall. Here, some of the prevailing works in rainfall prediction are briefed with their merits and issue that encouraged the formulation of the current work.

2.1 Literature Review

Among various studies that have focused on developing effective rainfall prediction techniques, a few are briefed in this section. Dr.S. Soundararajan, *et al.* [8] developed a Deep Convolutional Neural Network (Deep CNN) for predicting rainfall by considering the variation in the atmosphere and environmental factors, like humidity, temperature, and precipitation. This technique allowed the identification of variation/anomalies in the time-series data but the approach was not evaluated in comparison to other techniques. R. Venkatesh, *et al.* [12] proposed a Generative Adversarial Network (GAN) to forecast rainfall based on the rainfall information in India. This method successfully achieved high accuracy; however, it required high computational resources. Budiman, H. and Naparin, H., [22] developed a Backpropagation Neural Network (Backpropagation NN) for forecasting rainfall. This technique achieved high accuracy with minimal error, but the high performance achieved was with minimal input data. Chong, K.L. *et al.* [11] presented a CNN with Discrete Wavelet Transform (DWT) for predicting the time-series rainfall data. Though this scheme successfully captured the rainfall data patterns while forecasting or monthly, it failed to compute the uncertainty individually in various zones.

2.2 Challenges

The prevailing rainfall prediction schemes endure the following challenges.

- In [12], GAN was employed in forecasting rainfall monthly or daily, the main issue faced by the approach is that it failed to minimize the computational time endured during the training stage for satisfying the requirements in real-time scenarios.
- The Backpropagation Neural Network (NN) was proposed in [22] for forecasting rainfall, this technique though achieved high accuracy and it was unsuccessful in improving the efficiency of prediction using more input data.
- The main challenge encountered by CNN in [11] was that it did not consider the utilization of statistical methods, like MannKendall test, and box-cox transformation to address the issues faced due to the uncertainties, occurring because of variations in water regimes, temperature, and climate.

- Most of the prevailing techniques in rainfall prediction cannot identify the non-visible features in the data and suffer from high computational complexity. Further, the non-availability of rainfall data affects the performance efficiency.

3 Proposed CIOA-Deep LSTM for Rainfall Prediction

This section presents the devised CIOA-Deep LSTM for rainfall prediction. The developed CIOA-Deep LSTM is realized using the following phases. The primary step in the realization of the proposed rainfall prediction technique is the acquisition of input time series data from the dataset [23]. The accumulated data is then utilized in the extraction of various technical indicators, like Double Exponential Moving Average (DEMA), Adaptive Moving Average (AMA), Triple Exponential Moving Average Oscillator (TRIX), Moving Average Convergence Divergence (MACD), Money Flow Index (MFI), Price Channel (PC), Rate of Change (ROCP), and Time Series Forecast (TSF). From the multiple features determined, the best feature that precisely represents the input data are selected based on Tanimoto similarity and Dice similarity. Finally, rainfall prediction is carried out using the Deep LSTM [24], whose weight parameters are adjusted using CIOA [25]. The structural design of the devised CIOA-Deep LSTM is displayed in Fig. 1.

3.1 Data Gathering

In this work, rainfall is predicted based on the data containing the regular observations of the weather from numerous weather stations in Australia, and this time series data is acquired from a dataset R, which can be represented using the following expression,

$$R = \{R_1, R_2, .., R_i, ..., R_r\} \tag{1}$$

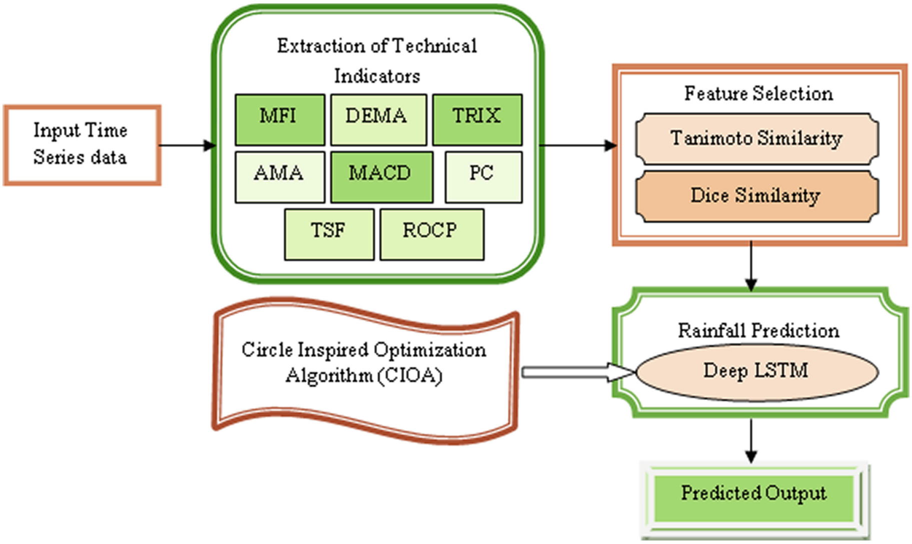


Fig. 1. Structural design of the devised CIOA-Deep LSTM for rainfall prediction

Here, r indicates the overall count of data contained in the dataset, and R_i represents the i^{th} data that is considered for further processing.

3.2 Extraction of Technical Indicators

Several technical indicators, such as DEMA, AMA, TRIX, MACD, MFI, PC, ROCP, and TSF are employed in the evaluation of the time-series data utilized in the proposed rainfall prediction scheme, and these indicators are determined for the input data R_i. The eight technical indicators utilized are briefed as follows.

i) DEMA: The DEMA indicator aims at providing a smoothed average with minimal lag in comparison to the straight Exponential Moving Average (EMA) and is computed based on the expression given below,

$$DEMA = (2 * EMA(a)) - (EMA(a)\, of\ EMA(a)) \tag{2}$$

Here, a denotes the period.

ii) AMA: The AMA indicator is sensitive to fluctuations in data, and it becomes highly sensitive in case of data change in a specific direction and is least sensitive in case of unstable data movements. It is represented by,

$$AMA = AMA(1) + b * (close - AMA(1)) \tag{3}$$

Here, *close* is the value of the rainfall at the period end, smoothing constant b is given by $b = \left[(Vi * (fc - sc)) + sc\right]$, where Vi indicates the user-defined quantification of trend strength or volatility, fc and sc denote the fastest and slowest smoothing constant $fc = 2\big/(fn + 1)$, and $sc = 2\big/(sn + 1)$, where fn signifies the fast period and sn denotes the slow period.

iii)TRIX: It is used to display the percentage rate change among two triple-smoothed EMAs and is the momentum indicator that keeps oscillating about zero, and is calculated using the following formula,

$$Trix = (EMA_{3a} - EMA_{3a-1})\big/EMA_{3a-1} \tag{4}$$

Here, EMA_{3a} and EMA_{3a-1} indicates the triple EMA at the current and previous periods.

iv) MACD: MACD is a technical indicator that gives the difference among two Moving Averages (MAs) of various lengths, and is expressed as,

$$MACD = F_{MA} - S_{MA} \tag{5}$$

Here, S_{MA} is the longer MA and F_{MA} represents the shorter MA.

v) MFI: This parameter utilizes both flow and volume of the rainfall to quantify the rainfall, and is computed based on the following formula,

$$MFI = 100 - \left(100\big/(1 + RR)\right) \tag{6}$$

Here, RR is the rainfall ratio.

vi) PC: The PC shows two bands, such as the upper band indicating the maximal value of rainfall in the last a periods and the lower band representing the lowest value of rainfall.
vii) ROCP: This indicator is also referred to as the momentum indicator and it is used to compare the present value of data to the past values and is calculated based on the following expression,

$$ROCP = (CR - Rf_a) - 1 \tag{7}$$

Here, CR indicates the current rainfall, and Rf_n is the rainfall values a years ago.
viii) TSF: It is also known as moving linear regression and is used to compute the present regression value of every bar based on the least square fit technique. TSF is calculated with the expression below,

$$y = fx + g \tag{8}$$

Here, f represents the slope and g is the interception.

All the extracted technical indicators are grouped to form the feature vector, which is formulated as,

$$M = \{m_1, m_2, ..., m_8\} \tag{9}$$

Here, m_1 is the DEMA, m_2 represents the AMA indicator, m_3 signifies TRIX, m_4 denotes MACD, m_5 symbolizes MFI, m_6 refers to PC, m_7 implies ROCP, and m_8 characterizes TSF. The feature vector thus produced is forwarded to the feature selection phase.

3.3 Feature Selection Using Tanimoto and Dice Similarity

The determined feature vector M comprises multiple features, and from these features, the best feature that precisely represents the input data are selected based on Tanimoto similarity and Dice similarity. The process of feature selection based on the Tanimoto similarity and Dice similarity is detailed in the succeeding sections.

Tanimoto Coefficient Similarity

The Tanimoto metric measures the similarity by taking on values between 0 and 1, with 1 being assigned in the case of identical vectors, and 0 being measured for non-identical vectors. The Tanimoto similarity is computed by,

$$M_{k \times u} = \frac{C \cdot D}{\|C\|^2 + \|D\|^2 - C \cdot D} \tag{10}$$

where, C and D indicates the candidate and class features, respectively. Once the Tanimoto coefficients are computed for all features, the top k valued features with high values are selected, with $k > s$, where $s \times u$ is the size of the input feature.

Dice Similarity

The optimal features are selected by subjecting the Tanimoto similarity-measured feature vector $M_{k \times u}$ to the Dice similarity coefficient measurement. The Dice similarity is a statistical technique used to quantify the similarity among two feature vectors. It effectively eliminates the redundant and irrelevant features in the data and is formulated as.

$$M_{k\times l} = 2 * (C \cap D) \Big/ (|C| + |D|) \tag{11}$$

Here, $l > u$, and top valued features are selected after computing the dice similarity of each feature, and the feature thus selected denoted by B is then employed in the rainfall prediction.

3.4 Rainfall Prediction Using the Developed CIOA-Deep LSTM

After preparing the rainfall data using the above-detailed techniques, the selected features B is then given to the Deep LSTM [24] for predicting the rainfall. The weights of the Deep LSTM are determined using the CIOA [25]. The process of rainfall prediction using the proposed CIOA-Deep LSTM is elucidated in this section.

Deep LSTM

The Deep-LSTM [24] is comprised of a stack of LSTM units, which capture the non-linear features of the input data. It also has the ability to achieve enhanced performance by exploring the various intrinsic features of time series data over a long duration. The optimal feature vector selected B is applied to the Deep LSTM for accomplishing rainfall prediction. The Deep LSTM schemes utilize the concept of memory cells and thereby, effectively handling the vanishing gradient issue. It comprises few internal contextual state cells that perform as short-term as well as long-term memory cells, and based on the state of these cells, the output varies. The Deep LSTM working totally relies on the memory cells. Consider that a node c_t in the Deep LSTM is provided with an input d_t from the input layer and prior hidden states e_{t-1} at a time t. The output of the Deep LSTM is given by,

$$e_t = \tanh(z_t)\Theta u_t \tag{12}$$

where, $z_t = c_t\Theta h_t + z_{t-1}\Theta v_t$, with z_t denoting the internal state at time t, Θ represents pointwise linear operator, h_t and u_t indicate the function of the input and output gates. The structural design of the Deep LSTM is explicated in Fig. 2.

CIOA for Weight Optimization of the Deep-LSTM

The weight parameters of the Deep-LSTM are adapted based on the CIOA [25], which is inspired by the commonly utilized properties of the trigonometric circle. The CIOA uses search agents that define the arc trajectories and are controlled by two distinct parameters, like the radius of the circumference and the user-determined angle. The CIOA offers high efficiency and robustness and is effective in addressing problems with multiple constraints. The algorithmic steps of the CIOA are listed below.

Step i) Initialization: The primary step in the CIOA is the initialization of the search agents, wherein all the search agents travel along the arc driven by two factors, like the

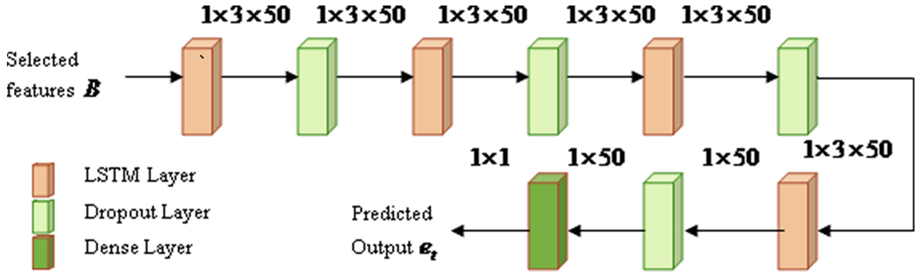


Fig. 2. Structural design of Deep LSTM

radius γ computed by the algorithm, and the user-defined angle φ. Consider a population of search agents to be represented as,

$$C = \{C_1, C_2, ..., C_s, .., C_D\} \tag{13}$$

Here, D denotes the total count of search agents, and every agent C_s is initialized with a radius given by,

$$\gamma_s = E_\gamma.s^2 \big/ D,\ 1 \le s \le D \tag{14}$$

Here, E_γ is a constant, formulated as $E_\gamma = \frac{1}{D}\sqrt{ub - lb}$, ub and lb indicate the maximum and minimal limits of the variable.

Step ii) Fitness function: The problem is formulated as a minimization issue and so the fitness parameter considered here is the MSE, which is given by,

$$MSE = \frac{1}{\alpha}\sum_{i=1}^{\alpha}\left(e_t^i - e_t^{i*}\right)^2 \tag{15}$$

where, e_t and e_t^* indicates the actual and the targeted output of the Deep LSTM. The agents are categorized in ranking based on the achieved solution quality.

Step iii) Upgrade search agents position: The location of the search agents relies on the categorization executed in the preceding iteration. The search agents are updated based on the expression given below.

$$Z_{2s}(q+1) = Z_{2s}(q) - W_1.\gamma_s.\sin(q.\varphi) + W_2.\gamma_s.\sin((q+1).\varphi) \tag{16}$$

$$Z_{2s-1}(q+1) = Z_{2s-1}(q) - W_3.\gamma_s.\cos(q.\varphi) + W_4.\gamma_s.\cos((q+1).\varphi) \tag{17}$$

wherein, q indicates the present iteration, $2s$ and $2s-1$ indicate the even and odd numbers. $W_i, i = 1\ to\ 4$ are arbitrary numbers following a uniform distribution with value in range [0, 1]. Once the search agents are updated, the value of the variable Z_s is checked to ensure the lower and upper limits are not exceeded, and if exceeded, the variable Z_s is assigned a value of the agent who has attained the optimal solution.

Step iv) Update radius: When a complete lap is covered by the search agent after s iterations, the new radius value $\vec{\gamma}_{new}$ is computed using the following expression,

$$\vec{\gamma}_{new} = \vec{\gamma}0.99 \tag{18}$$

Step v) Update variable range: A variable G_{ls} is used to restrict the agents to the most hopeful regions in the search space, thus performing an exclusive local search. The exclusive local search begins at the q^{th} iteration, while the proportion of s to the total iteration count is higher than G_{ls}. Correspondingly, the lower and upper limits are computed as,

$$lb_{1s} = Z_{sbest} - \frac{ub - lb}{10000} \tag{19}$$

$$ub_{1s} = Z_{sbest} + \frac{ub - lb}{10000} \tag{20}$$

where, Z_{sbest} indicates the variable that attains the optimal solution in s^{th} dimension.

Step vi) Re-evaluate fitness: After upgrading the location of the agents, the fitness of the agents is computed with Eq. (15), and the agent with minimal fitness is identified as the best solution.

Step vii) Terminate: The process is reiteration till the maximal iteration count is attained. Algorithm 1 depicts the pseudocode of the CIOA.

Algorithm 1. Pseudocode of CIOA

```
Begin
Specify $\phi$ and $G_{ls}$
    Initiate radii $\vec{\gamma}$ based on equation (14) and allocate arbitrary values to design variables
    Compute fitness using equation (15)
While1 $(s \leq G_{ls} \times s_{\max})$
        Categorize search agents based on the obtained solution's quality
        Modify agent position using equations (16) ad (17)
        Check whether the design variable exceeds the maximal and minimal bounds
If $s$ is a multiple of $round(360/\phi)$
            Modify $\vec{\gamma}$ using equation (18)
End If
    End While 1
    Allocate the location of the agent with the best solution to all agents
    Modify variable range with equations (19) and (20)
While 2 ( $s \leq s_{\max}$ )
        Repeat While1, with the new variable range
EndWhile 2
Return the best solution
End
```

Thus, by the weight optimization of the Deep LSTM based on the CIOA, the devised CIOA-Deep LSTM effectively performs rainfall prediction with minimal error.

4 Results and Discussion

The developed CIOA-Deep LSTM is examined for its efficacy considering various parameters in comparison to the prevailing rainfall prediction methods, and this is detailed in this section.

4.1 Experimental Set-Up

The proposed CIOA-Deep LSTM is realized on a PC with Windows 10, 8 GB RAM, and Intel i5 processor in a Python environment.

4.2 Dataset Description

The experimentation of the developed CIOA-LSTM is accomplished using the Australian weather dataset [23], which comprises nearly 10 years of everyday observations of weather at different locations all over Australia.

4.3 Evaluation Measures

The efficacy of the proposed prediction model is investigated by considering metrics, like Mean Square Error (MSE), Root Mean Square Error (RMSE), and Relative Absolute Error (RAE).

a) ***MSE***: MSE parameter denotes the mean of the square of the variation between the targeted and actual output of the Deep LSTM and is computed using Eq. (15).
b) ***RMSE:*** This metric is computed by determining the square root of MSE, and is represented as,

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{\alpha}\sum_{i=1}^{\alpha}\left(e_t^i - e_t^{i*}\right)^2} \tag{21}$$

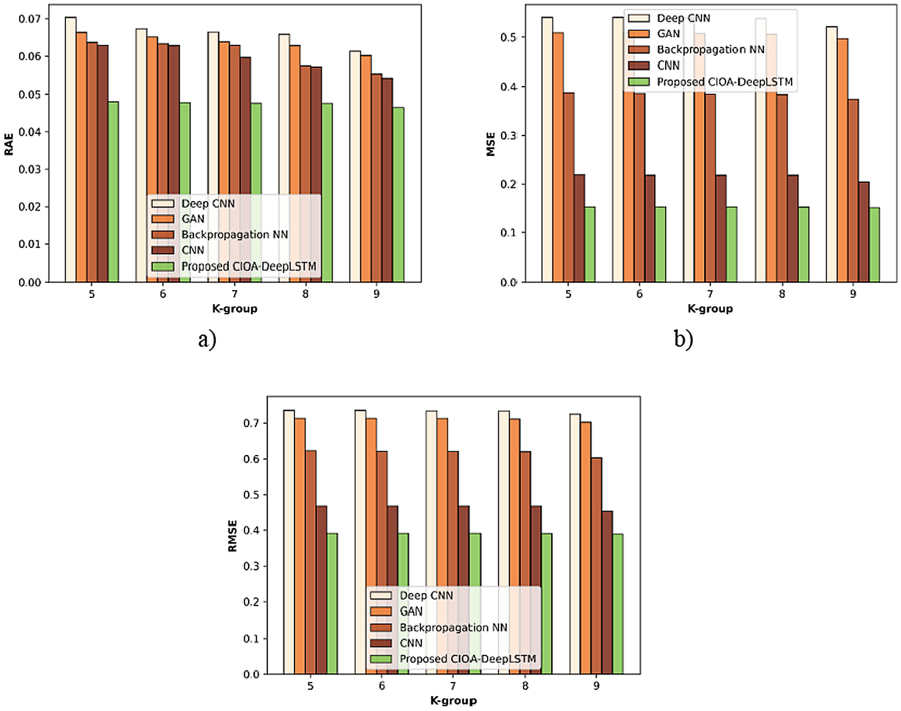
c) ***RAE:*** It measures the ratio of residual or mean error to the error generated by the Deep LSTM, and is quantified by,

$$RAE = \frac{\sum_{i=1}^{\alpha}\left|e_t^i - e_t^{i*}\right|}{\sum_{i=1}^{\alpha}\left|e_t^i - \overline{e}\right|} \tag{22}$$

where, $\overline{e} = \sum_{i=1}^{\alpha} e_t^i$

4.4 Comparative Techniques

The performance of the presented prediction model is examined considering various rainfall estimation schemes, like Deep CNN [8], GAN [12], Backpropagation NN [22], and CNN [11].



a)

b)

c)

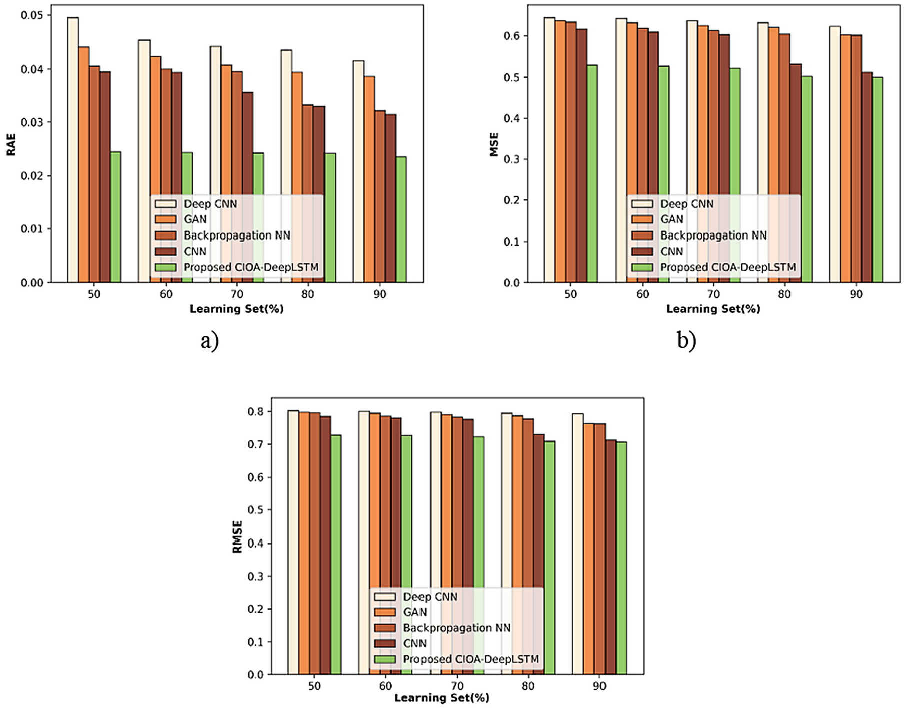
Fig. 3. Assessment considering k-group based on a) RAE, b) MSE, and c) RMSE

4.5 Comparative Assessment

The comparison of the effectiveness of the prediction schemes considering k-group and learning set based on different parameters is depicted in this section.

Evaluation Based on K-Group

The developed CIOA-Deep LSTM is evaluated considering k-group and this is illustrated in Fig. 3. Figure 3a) shows the evaluation of the devised CIOA-Deep LSTM based on RAE. With k-group of 4, the RAE value attained by the rainfall prediction schemes, like Deep CNN, GAN, Backpropagation NN, CNN, and the presented CIOA-Deep LSTM is 0.070, 0.066, 0.064, 0.063, and 0.048, correspondingly. In Fig. 4b, the MSE-oriented assessment of the developed CIOA-Deep LSTM is illustrated. The various prediction models computed MSE of 0.541for Deep CNN, 0.509 for GAN, 0.386 for Backpropagation NN, 0.219 for CNN, and 0.153 for the devised CIOA-Deep LSTM, with k-group of 5. Figure 3c) presents the analysis of the current work based on RMSE. The value of RMSE attained is 0.734, 0.713, 0.620, 0.467, and 0.391, corresponding to Deep CNN, GAN, Backpropagation NN, CNN, and the presented CIOA-Deep LSTM, with k-group of 7.



a) b) c)

Fig. 4. Valuation based on learning set considering a) RAE, b) MSE, and c) RMSE

Assessment Considering Learning Set
Figure 4 displays the examination of the developed CIOA-Deep LSTM considering different learning sets. Figure 4a) illustrates the RAE-oriented evaluation of the current work. The RAE value achieved by the devised CIOA-Deep LSTM is 0.024, with 60% learning set, while the other techniques, such as Deep CNN, GAN, Backpropagation NN, and CNN attained RAE of 0.045, 0.042, 0.040, and 0.039. In Fig. 4b), the assessment of the proposed prediction scheme concerning MSE is presented. The techniques, like Deep CNN, GAN, Backpropagation NN, CNN, and the devised CIOA-Deep LSTM measured MSE of 0.636, 0.624, 0.612, 0.602, and 0.522, with 70% learning set. The RMSE-oriented assessment of the current work considering the learning set is portrayed in Fig. 4c). The RMSE value quantified is 0.794 for Deep CNN, 0.787 for GAN, 0.776 for Backpropagation NN, 0.730 for CNN, and 0.709 for the current work for 80% learning set.

4.6 Comparative Discussion

The comparative discussion of the current work is accomplished in this section, in which the performance of the current work is examined based on various metrics, like RAE, MSE, and MSE in comparison to the prevailing rainfall forecasting schemes. Table 1

Table 1. Comparative discussion of the developed CIOA-Deep LSTM for rainfall prediction

Variation	*Metric*	*Deep CNN*	*GAN*	*Backpropagation NN*	*CNN*	*Proposed CIOA-Deep LSTM*
K-group	*RAE*	0.061	0.060	0.055	0.054	0.046
	MSE	0.521	0.497	0.375	0.205	**0.151**
	RMSE	0.726	0.703	0.603	0.453	**0.389**
Learning set	*RAE*	0.042	0.039	0.032	0.031	**0.023**
	MSE	0.621	0.601	0.600	0.513	0.500
	RMSE	0.794	0.763	0.762	0.713	0.707

shows the comparison of the presented technique, with values depicted relating to k-group of 9 and learning set of 90%. The current work is observed to have attained a low value of RAE at 0.023 MSE of 0.151, and RMSE of 0.389 owing to the utilization of CIOA-Deep LSTM for predicting rainfall.

5 Conclusion

This work presents the potential of an optimized DL-based model, named CIOA-Deep LSTM for predicting rainfall based on the time-series rainfall data. DL approaches have the capability to learn the non-linear features in the time-series information and hence can be effectively utilized in the process of rainfall prediction. Here, rainfall prediction is performed by utilizing a Deep LSTM, whose weight parameters are optimized by the CIOA. The input rainfall data is subjected to the extraction of technical indicators, during which a total of eight indicators were extracted. The indicators that represent the most relevant information in the input are selected using Tanimoto and Dice similarity. The selected features are applied to the Deep LSTM for rainfall prediction, and it is trained by the designed optimization algorithm. The effectiveness of developed CIOA-Deep LSTM is examined based on parameters, like RAE, MSE, and RMSE. Experimental results show that the devised CIOA-Deep LSTM reveals enhanced performance by attaining a minimal value of RAE at 0.023 MSE of 0.151, and RMSE of 0.389. However, the performance of the devised approach is evaluated with a single dataset. In the future, the performance will be evaluated by considering more complex datasets. Also, the efficacy of the developed technique can be improved by considering hybrid DL approaches to close the gap between the targeted and attained outputs.

References

1. Yucel, I., Onen, A., Yilmaz, K.K., Gochis, D.J.: Calibration and evaluation of a flood forecasting system: Utility of numerical weather prediction model, data assimilation and satellite-based rainfall. Journal of Hydrology523, 49–66 (2015).
2. Czarnecka, M., Nidzgorska-Lencewicz, J.: Impact of weather conditions on winter and summer air quality. International Agrophysics25(1) (2011).

3. Mokrani, H., Lounas, R., Bennai, M.T., Salhi, D.E., Djerbi, R.: Air quality monitoring using iot: A survey. In Proceedings of 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), IEEE 127–134 (2019).
4. Barrera-Animas, A.Y., Oyedele, L.O., Bilal, M., Akinosho, T.D., Delgado, J.M.D., Akanbi, L.A.: Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. Machine Learning with Applications7, 100204 (2022).
5. Aswini, K.R.N., Raghavan, S.V., Sreekanth, N.P., Sree R.P.M.: For Effective, Earlier and Simplified Diagnosis of Retinopathy of Prematurity (RoP), a Probe through Digital Image Processing Algorithm in B-Scan," Medico-Legal Update, vol. 20(3, pp. 105–109, (2020).
6. Aswini, K.R.N., Vijayaraghavan, S.: Denoising of Ultrasonic B–Scan Retinal Images for ealier Detection of Retinopathy of Prematurity (RoP)," New Frontiers in Communication and Intelligent System, pp. 661–670, (2021).
7. Aguasca-Colomo, R., Castellanos-Nieves, D., Méndez, M.: Comparative analysis of rainfall prediction models using machine learning in islands with complex orography: Tenerife Island. Applied Sciences9(22), 4931 (2019).
8. Soundararajan,S., Chinnuswamy, Visuwasam, L.M.M.,Pushparathi,V.P.G., Sudha,M.: A Novel Deep Learning Framework For Rainfall Prediction In Weather Forecasting. Turkish Journal of Computer and Mathematics Education (TURCOMAT)12(11), 2685–2692 (2021).
9. Pham, Q.B., Abba, S.I., Usman, A.G., Linh, N.T.T., Gupta, V., Malik, A., Costache, R., Vo, N.D., Tri, D.Q.: Potential of hybrid data-intelligence algorithms for multi-station modelling of rainfall. Water Resources Management33(15), 5067–5087 (2019).
10. Jhong, B.C., Huang, J., Tung, C.P.: Spatial assessment of climate risk for investigating climate adaptation strategies by evaluating spatial-temporal variability of extreme precipitation. Water Resources Management33(10), 3377–3400 (2019).
11. Chong, K.L., Lai, S.H., Yao, Y., Ahmed, A.N., Jaafar, W.Z.W., El-Shafie, A.: Performance enhancement model for rainfall forecasting utilizing integrated wavelet-convolutional neural network. Water Resources Management34, 2371–2387 (2020).
12. Venkatesh,R., Balasubramanian,C., Kaliappan,M.: Rainfall prediction using generative adversarial networks with convolution neural network. Soft Computing25(6), 4725–4738 (2021).
13. Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., and Bacanin, N.: Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection, Mathematics, 10(13): 1–30, (2022).
14. Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K and Abouhawwash, M.: Modified firefly algorithm for workflow scheduling in cloud-edge environment, Neural Computing and Applications, 34: 9043–9068, (2022).
15. Zivkovic, M., Petrovic, A., Venkatachalam, K., Strumberger, I., Jassim, H., and Bacanin, N.S.: Novel Chaotic Best Firefly Algorithm: COVID-19 Fake News Detection Application, Advances in Swarm Intelligence, 1054: 285–305, (2022).
16. Bacanin, N., Zivkovic, M., Sarac, M., Petrovic, A., Strumberger, I., Antonijevic, M., Petrovic, A., and Venkatachalam, K.: A Novel Multiswarm Firefly Algorithm: An Application for Plant Classification, In the proceeding of International Conference on Intelligent and Fuzzy Systems, 504: 1007–1016, (2022).
17. Bacanin, N., Stoean, C., Zivkovic, M., Jovanovic, D., Antonijevic, M and Mladenovic, D.: Multi-Swarm Algorithm for Extreme Learning Machine Optimization, Sensors, 22(11): 1–34, (2022).
18. Bacanin, N., Antonijevic, M., Bezdan, T., Zivkovic, M., Venkatachalam, K and Malebary, S.: Energy efficient offloading mechanism using particle swarm optimization in 5G enabled edge nodes, Cluster Computing, (2022).

19. Bacanin, N., Zivkovic, M., Al-Turjman, F., Venkatachalam, K., Trojovský, P., Strumberger, I and Bezdan, T.: Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application, Scientific Reports, 12, (2022).
20. Salb, M., Jovanovic, L., Zivkovic, M., Tuba, E., Elsadai, A., and Bacanin, N.: Training Logistic Regression Model by Enhanced Moth Flame Optimizer for Spam Email Classification, Computer Networks and Inventive Communication Technologies, 141: 753–768, (2022).
21. Khan, M.I., Maity, R.: Hybrid deep learning approach for multi-step-ahead daily rainfall prediction using GCM simulations. IEEE Access8, 52774–52784 (2020).
22. Budiman,H., Naparin, H.: Rainfall Prediction Using Backpropagation neural Network Algorithm. Journal of Health Informatics Management, Education, and Law2(1), 22–29 (2021).
23. Australian weather dataset available at, "https://www.kaggle.com/jsphyg/weather-dataset-rattle-package" accessed on September 2022.
24. Majhi, B., Naidu, D., Mishra, A.P., Satapathy, S.C.: Improved prediction of daily pan evaporation using Deep-LSTM model. Neural Computing and Applications32(12), 7823–7838 (2020).
25. de Souza, O.A.P., Miguel, L.F.F.: CIOA: Circle-Inspired Optimization Algorithm, an algorithm for engineering optimization. SoftwareX19, 101192 (2022).