Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 47 No. 03
Aug 2026

Enhancement of Rainfall Prediction Model using Convolutional Neural Network Optimized by Chameleon Particle Swarm Optimization for Satellite Communication Application

Ajadi A. S.[1], Adeyemo Z. K.[2], Semire F. A.[2], Akande D. O.[2*].

[1] Department of Electrical Engineering, Ajayi Crowther University, Oyo, Nigeria.
[2] Department of Electronic and Electrical Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

Abstract

Accurate rainfall prediction is critical for dependable wireless and satellite communications, especially in tropical areas where rain attenuation significantly degrades the reliability of these systems. While Convolutional Neural Networks (CNN) have proven effective in rainfall modelling due to rainfall's non-linear characteristics, achieving optimal performance requires careful tuning of CNN's hyperparameters. This research introduces a Chameleon Particle Swarm Optimization (CNN-CPSO) aimed at the automatic optimization of CNN hyperparameters with improved rainfall prediction capability. The simulation was performed using monthly rainfall data from Modern-Era Retrospective Analysis for Research and Applications version 2 (MERRA-2) obtained from the National Aeronautics and Space Administration (NASA), 1980-2020. Nigerian Meteorological Agency (NiMet) observation data were used for model evaluation. The rainfall data was pre-processed and normalized, and then partitioned into training, validation, and testing sets. The CNN model was implemented in the deep learning paradigm, while the Chameleon Particle Swarm Optimization (CPSO) technique was used to optimize some of the CNN's key hyperparameters, including learning rate, batch size, dropout rate, number of convolutional layers, and filters. The performance of the developed model was determined using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2).

Compared to the conventional CNN and CNN optimized with the standard Particle Swarm Optimization (CNN-PSO), the proposed CNN-CPSO showed improved prediction performance with a lower MSE of 1.12, RMSE of 1.06, and MAE of 0.89, an R^2 of 0.94, and a prediction accuracy of 93.30%. This compared to 87.90% for CNN-PSO and 84.60% for the conventional CNN, respectively. Therefore, the CNN-CPSO is recommended for use in weather forecasting, rain fade prediction and mitigation in satellite communications, hydrological modeling, and climate change studies.

Keywords: CNN-CPSO, Rainfall Prediction, Optimization, Rain Fade Mitigation, Ka-band Satellite Communication.

1. Introduction

Rainfall prediction has long been identified as a challenging issue in environmental and communication engineering because of the underlying processes' nonlinearity, randomness, and complexity. Rainfall in tropical countries such as Nigeria exhibits significant temporal and spatial variability, making accurate prediction using conventional statistical models particularly challenging [1]. Accurate rainfall prediction is critical to Ka-band satellite communication systems because rain attenuation is the main climatic impairment that causes signal fading and outage [2][3]. Artificial intelligence has significantly transformed rainfall prediction in the past years [4]. Deep learning models, specifically Convolutional Neural Networks, were proven to be more efficient in retrieving the underlying spatial and temporal features from meteorological datasets. CNNs automatically learn hierarchical representations from rainfall time series, which enables them to achieve much better forecasting performance than require engineering features to be defined by human expert [5][6][7].

Although the performance of a convolutional neural network has many advantages, its efficiency is highly dependent on various factors, including the choice of a learning rate, kernel size, filters number, dropout rate, batch size, and network depth. The selection of these parameters requires extensive computation to arrive at optimal architecture, which is not only time-consuming but also prone to yielding inaccurate models. As a result, there has been a proliferation of research studies aimed at integrating optimization algorithms into the development process of deep neural networks [8][9].

Particle Swarm Optimization has been widely adopted for its easiness and ability to find global optimum solutions. Conventional PSO, however, has several drawbacks like premature convergence, low exploration capability and poor performance for high dimensional problems, which may undermine the performance of CNN models [10][11].

In order to solve this problem, this work developed an Enhanced Convolutional Neural Network optimized using Chameleon Particle Swarm Optimization (CNN–CPSO). CPSO utilizes adaptive movement, dynamic exploration, and exploitation mechanisms inspired by the hunting behaviour of chameleons to achieve global optima while avoiding local optima convergence [12][13].

The combination of CNN and CPSO is an efficient prediction system that can automatically recognize best network structures for maximizing rainfall prediction capability with reduced computational complexity. The proposed method can be applied to a number of fields that require accurate estimation of rainfall, including adaptive satellite communication, hydrology, flood management, agriculture, and climate studies. The rest of the paper is divided into sections as described below. The proposed method is discussed in detail in Section II. The materials and methods used in the study are presented in Section III. The simulation results are discussed in Section IV. Finally, a brief conclusion is drawn in section V.

2. Related Works

Hnin and Jeenanunta (14) presents an approach to optimize the Support Vector Regression (SVR) for short-term load forecasting with use of particle swarm optimization method. The work proposed PSO to tune SVR parameters, penalty, epsilon and kernel parameter, considering Thailand dataset. It shows that PSO improved SVR to a higher prediction value over classical SVR, i.e., better prediction accuracy, indicated by smaller MAPE values than usual SVM value. However, this work pointed out that usual PSO algorithm might converge to local minima; it has premature convergence problem leading it away to the global optimum, thus suggesting that hybridizing other optimization techniques would benefit the global exploration ability.

In a study that improved rainfall forecasting, Umamaheswari (15) developed a hybrid model of LSTM-PSO that applied Particle Swarm Optimization to optimize the LSTM hyperparameters, which resulted in the LSTM-PSO model yielding high precision and prediction errors lower compared to the standard LSTM, Bi-LSTM and GRU models, although the problem did not analyses possible failures in PSO such as premature convergence or lower ability to make global searches.

Hameed et al. (16) proposed an LLM-powered Particle Swarm Optimization (LLM-PSO) that leverages an LLM to tune hyperparameters for deep learning applications. By combining ChatGPT-3.5 and Llama3 with the Particle Swarm Optimization technique, the work increases convergence speed and reduces costs from 20-60% for LSTM and CNN models. Prediction accuracies, however, remained consistent throughout the experiment. The drawback of the solution is the dependence on the quality of prompts written by the LLM and a lack of external testing on more extensive data.

Abdishakur et al. (17) recommended hybridizing a CNN model and a GBM (GBM-CNN hybrid) model on a basis of time series temperature and humidity to rainfall forecasts. Authors noted that the combined models perform effectively, it outperforms CNN, individually. But the model has used limited meteor variables to forecasts and also did not consider using a metaheuristic optimization to improve its robustness and performance

Satheesh et al. (18) formulated gamma regression and CNN models and performed daily rainfall forecasting in Northern Tropical Africa with use of tropical wave predictors. Their paper concluded that the results from their CNN model had higher forecasting accuracy and demonstrated improved probabilistic performances compared with other forecast methods based on traditional numerical prediction models and climatology. But the work was solely focused on only one type of predictors, that is, tropical wave-only predictors, and did not consider usage of metaheuristic optimization algorithms for fine-tuned optimization of parameters in CNN models and better forecasting in comparison to methods already explored.

In the review paper of Liu et al. (19), the use of deep learning models for weather forecasting had been investigated. Several kinds of deep learning models, such as CNN, RNN, Transformer, GNN, and hybrid structure models, had been introduced, and these models have been tested and improved through extensive applications for prediction. It was reported that the integrated hybrid deep learning model tended to outperform an individual model because it could effectively learn spatiotemporal features. Nevertheless, the long-range dependence learning capability and accuracy degradation with longer time scales still remained significant issues for CNNs. Hybridization and optimized models were required to achieve a better performance.

Although CNN-based deep learning models and PSO-based optimization techniques have significantly improved rainfall prediction accuracy, existing studies largely rely on manually selected CNN hyperparameters or conventional PSO, which suffers from premature convergence, limited exploration capability, and computational inefficiency. Furthermore, no previous study has integrated Chameleon Particle Swarm Optimization (CPSO) with CNN to optimize hyperparameters for monthly rainfall prediction using the MERRA-2 dataset validated with Nigerian Meteorological Agency (NiMet) observations. This study tries to bridge the mentioned research gap by introducing a CNN–CPSO framework that enhances prediction accuracy, convergence, and computational efficiency, with potential application in Ka-band satellite communication.

3. Methodology

This paper developed enhanced rainfall prediction model which consists of a convolutional neural network (CNN) and chameleon particle swarm optimization (CPSO) algorithms. The CNN hyperparameters were optimized using the CPSO algorithm to better the rainfall forecasting accuracy. The model was trained using historical rainfall data from MERRA-2 and validated using the Nigerian Meteorological Agency (NiMet) observations. The proposed rainfall prediction model comprises four main stages: data preprocessing, convolutional neural network, hyperparameter optimization, and rainfall prediction. Figure 1 below shows the general design of the CNN–CPSO rainfall prediction model.

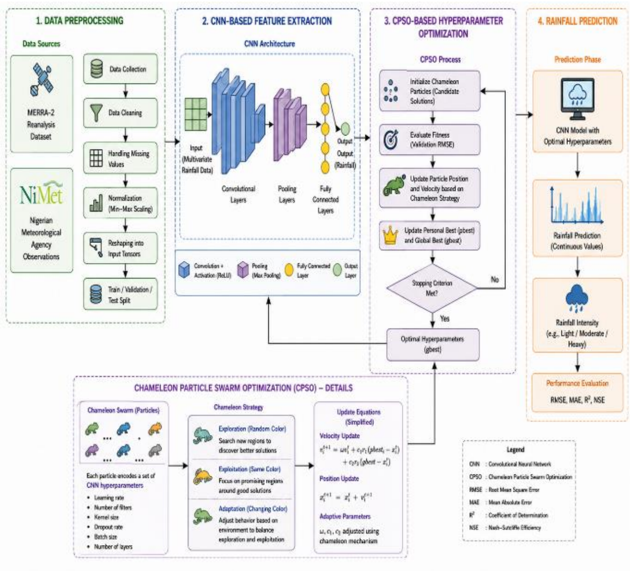

Figure 1: Overall Architecture of the developed CNN–CPSO Framework for Rainfall Prediction

A. Data Preprocessing

The rainfall dataset was initially processed and prepared to increase model performance and reduce inconsistencies. The missing values were filled using an interpolation algorithm, while the rainfall measurements were Standardized using the Minimum–Maximum algorithm to scale the data as presented in equation (1):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Where X_{norm} is the standardized rainfall value.
X represents the raw rainfall measurement obtained from the dataset, X_{min} and X_{max} denote the minimum and maximum rainfall values, respectively,

The normalized dataset was subsequently divided into training, validation, and test sets in a 70:15:15 ratio.

B. Formulation of CNN-CPSO Algorithm

A CNN-CPSO algorithm was formulated for this paper to select the best combination of CNN weights, number of layers, filter size and batch size. The general formulation of an optimal weight determination problem that was used in this work is expressed in equation (2)

$$\min_{Pbest, Gbest, W_f^d} \emptyset(y(W_f^d))$$

Subject to:

$$C1:\ 0 \le (wt, Obj, Pbest, Gbest, W^d) \le 1, \quad (2)$$
$$W^d \in W_e$$
$$C2:\ Obj = \begin{cases} 1 & if\ Obj \le \overline{Obj} \\ 0 & otherwise\ Obj > \overline{Obj} \end{cases}$$

where $wt \in R^n$ is the vectors of randomized weights. $\overline{Obj}$ is the mean square error for Obj.

The entire state vector is denoted as $y = [w_t]$, where s represents the set of CNN weights. The optimization problem is defined over the weight horizon $W_e = [W_o^d, W_f^d]$, where W_o^d represent the initial weights of y and W_f^d represent the final weights of y with W_f^d representing the optimal weight obtained in Algorithm 1. The local and global best position weight-dependent control variables, $P_{best}, G_{best} \in \mathbb{R}^n$, together with W_f^d, are the decision variables. The objective is to determine the optimal decision variables that minimize the objective function. The search space is constrained by error fitness and weight requirements. Two constraints are considered: C_1, which ensures weight values remain within $[0, 1]$, and C_2, which assigns a value of 1 to optimal weights and 0 to all other weights.

C. Solution to the formulated problem

The solution to the formulated problem was addressed by using Chameleon Particle Swarm Optimization (CPSO) based Convolution Neural Network demonstrated in Algorithm 1 which defined the standard CNN and CPSO respectively.

Algorithm 1: Chameleon Particle Swarm Optimization

Begin

1. Initialize parameters: Generate a random population of size $\boldsymbol{N}$. Define the PSO control parameters: inertia weight bounds $\boldsymbol{\omega_{\min}}, \boldsymbol{\omega_{\max}}$, and acceleration coefficients $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$.
2. Initialize particles for each particle, define its position vector $\boldsymbol{x_j}$ and velocity vector $\boldsymbol{v_j}$, representing the weight set $\boldsymbol{w_T} = [\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_n}]$.
3. Let $\boldsymbol{k = 1}$.
4. Compute the fitness $\boldsymbol{F_{ij}(t) = f\left(\vec{x}_{ij}(t)\right)}$ for all particles using the mean squared error:
$$\boldsymbol{\vec{x}_{ij}(t) = \frac{1}{n}\sum_{i=1}^{n}(p_o - p_p)^2}$$
where $\boldsymbol{p_o}$ is the observed value and $\boldsymbol{p_p}$ is the predicted value. Identify the index $\boldsymbol{b}$ of the particle with the best fitness.
5. Initialize personal best $\boldsymbol{Pbest_{ij}(t) = \vec{x}_{ij}(t)}$ and global best $\boldsymbol{Gbest_{ij} = \vec{x}_{bj}(t)}$.
6. Update the inertia weight using the chameleon-inspired formula
$$\boldsymbol{\omega = \left(1 - \frac{t}{T}\right)^{\left(\rho\sqrt{\frac{t}{T}}\right)}}$$
7. Update velocity using
$$\boldsymbol{\vec{v}_{ij}(t+1) = \omega\vec{v}_{ij}(t) + c_1 r_1\left(Pbest - \vec{x}_{ij}(t)\right) + c_2 r_2\left(Gbest - \vec{x}_{ij}(t)\right)}$$
If $\boldsymbol{r^i \geq p_p}$, update position using:
$$\boldsymbol{x_{t+1}^{(i,j)} = x_t^{(i,j)} + p_1\left(P_t^{(i,j)} - G_t^j\right)r_i + p_2\left(G_t^j - x_t^{(i,j)}\right)r_2}$$
8. Compute the new fitness $\boldsymbol{F_{ij}(t+1) = f\left(\vec{x}_{ij}(t+1)\right)}$ and determine the index $\boldsymbol{b_1}$ of the new best particle.
9. Update personal best positions for each particle, if $\boldsymbol{F_{ij}(t+1) < F_{ij}(t)}$, then set $\boldsymbol{Pbest_{ij}(t+1) = \vec{x}_{ij}(t+1)}$; otherwise, retain $\boldsymbol{Pbest_{ij}(t)}$.
10. If $\boldsymbol{F_{bj}(t+1) < F_{bj}(t)}$, update $\boldsymbol{Gbest_j(t+1) = Pbest_{bj}(t+1)}$ and set $\boldsymbol{b = b_1}$; otherwise, retain $\boldsymbol{Gbest_j(t)}$.
11. If $\boldsymbol{k} <$ Max_iter, increment $\boldsymbol{k = k+1}$ and return to Step 2. Otherwise, proceed to Step 12.
12. Output the optimal solution as the final global best position:
$$\boldsymbol{Gbest_{bj} = \vec{x}_{bj}(t)}$$

End

D. Optimization of Hyper-parameter

Pre-trained CNN architectures have limited flexibility because many hyperparameters are fixed. This study integrates the Chameleon Particle Swarm Optimization (CPSO) algorithm with a CNN classifier to automatically optimize key hyperparameters, including batch size, dropout rate, the number of convolutional layers, filter size, and convolutional filters. Figure 2 shows the CNN–CPSO framework, where each CPSO particle represents a candidate hyperparameter set evaluated through CNN training. The optimization iterates until all particles are assessed across successive generations. Using 10 particles and 10 iterations, CPSO performed 100 CNN training runs.

E. Learning Phase

During the learning phase, the CNN model classified rainfall rate data through feature extraction and fine-tuning of the ResNet architecture. The convolutional layers were retained for feature extraction, while the classifier was replaced with fully connected layers comprising a flatten layer, batch normalization, dropout, and two dense layers. The first dense layer used the ReLU activation function, and the second employed a four-class SoftMax output. After initial training, the last two convolutional layers were reactivated and fine-tuned with the classifier. The trained CNN–CPSO model then generated the final rainfall rate predictions using SoftMax posterior probabilities.

F. *CNN-CPSO Rainfall Prediction Model*

The developed CNN-CPSO is the prediction model that extracts the hidden spatial and temporal rainfall features from the historical rainfall sequences. Unlike traditional machine learning algorithms, the CNN-CPSO can automatically learn the features of the rainfall through consecutive convolution layers. For an input rainfall sequence, expressed as:

$$\text{x(t)} = [R(t-2), R(t-1), R(t)], \quad (2)$$

the developed CNN-CPSO predicts the future rainfall

$\hat{R}(t+1) = f\theta(x(t))$,

Where $f\theta$ $(\cdot)$ denotes the nonlinear mapping learned by the CNN, θ represents the trainable network parameters.

The flowchart of the developed CNN-CPSO is presented in Figure 2.

G. Model Performance Evaluation

The efficiency of the developed CNN–CPSO rainfall prediction model was computed using statistical performance indicators such as MSE, RMSE, MAE, MAPE, and R^2 to determine the accuracy, precision, and prediction capability of the proposed model.

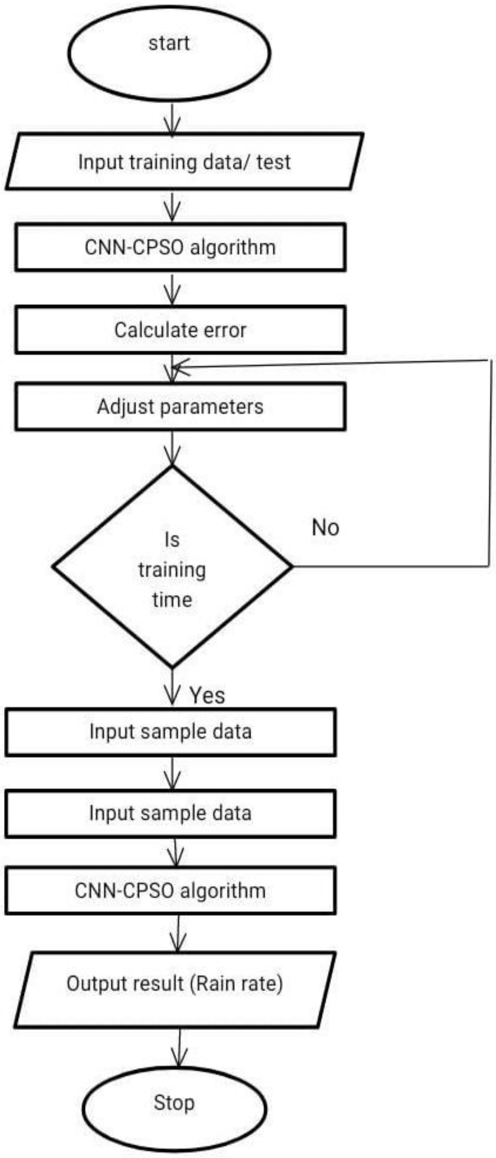

Figure 2: Flowchart of the developed CNN-CPSO Rainfall Prediction Model

The developed CNN–CPSO rainfall prediction model therefore provides an intelligent rainfall prediction framework that combines deep feature extraction with adaptive hyperparameter optimization to improve convergence, reduce prediction error, and enhance rainfall estimation accuracy for dynamic rain fade mitigation in Ka-band satellite communication systems and related environmental applications.

4. Results and Discussion

The outcomes of the CNN-CPSO-based rainfall prediction model evaluation are illustrated in this section. The simulations were carried out using MATLAB R2024a. In this regard, the CNN was designed using the Deep Learning Toolbox, while the CPSO paradigm was applied to optimize the hyperparameters. The performance of the CNN-CPSO was estimated using the MSE, RMSE, MAE, MAPE, and R^2. For comparison purposes, the CNN-based model and conventional approaches were chosen as the benchmark models. The obtained results demonstrate that the CNN-CPSO achieves better performance than the CNN in terms of accelerated convergence, reduced prediction errors, and improved prediction accuracy, which makes it a promising tool for real-time rainfall prediction and satellite communication applications.

A. Optimized CNN Hyperparameter Selection

The CPSO algorithm has been improved to optimize the key parameters, including the convolution filter size, the number of filters, the dropout rate, the batch size, and the learning rate, as shown in Table 1.

Table 1: Optimized CNN Hyperparameters Obtained Using CPSO

Parameter	Conventional CNN	CNN–CPSO Optimized Value
Number of convolution layers	2	3
Size of Filter	3 × 3	5 × 5
Filters Number	32	64
Rate of Learning	0.001	0.0005
Batch size	64	32
Dropout rate	0.50	0.25
Optimizer	Adam	Adaptive Adam

The optimized parameters allowed the developed model to achieve faster convergence and improved generalization capability.

B. Training and Convergence Performance

Figure 3 presents the training and validation loss curves of the CNN–CPSO model. It can be seen that the loss value of epochs decreases rapidly during the initial training process and then gradually converges, which means that CNN weights have been well optimized. The improved convergence behaviour is attributed to the adaptive search mechanism of CPSO, which prevents premature convergence and enables the discovery of better CNN configurations.

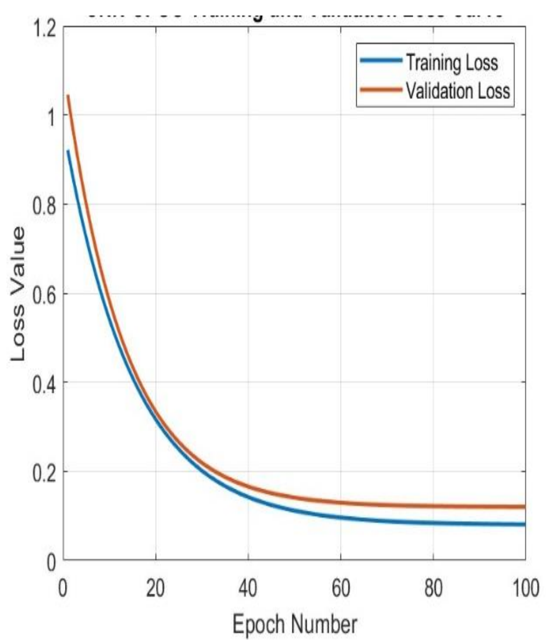

Figure 3: Training and Validation Loss Curve of CNN–CPSO

The developed CNN–CPSO training and validation losses declined steadily from epoch 50 to epoch 100. The training loss reduced from 0.120 to 0.075 while the validation loss reduced from 0.145 to 0.110. The small learning gap of 0.03–0.04 indicates that the CNN–CPSO model had good generalization performance with minimal overfitting. The CNN–CPSO training and validation curves flattened out after epoch 80, which showed that the CNN–CPSO had converged. This confirmed that the CPSO had optimized the CNN hyperparameters for stable rainfall prediction.

C. Comparative Prediction Performance

Figure 4 to 7 represents the performance evaluation of rainfall developed prediction models based on MSE, RMSE, MAE, and R^2. From the figures, it is clear that CNN–CPSO algorithm has achieved better results than others in all performance measures.

Figure 4 shows the comparison of the Mean Squared Error (MSE) of the CNN, CNN–PSO and CNN–CPSO models. It can be observed that the conventional CNN had the maximum MSE value of 2.84, which suggests that prediction errors using this algorithm were relatively large. With the improvement of PSO, the MSE reduced to 1.79, which was 36.97% lower than that of the CNN. The lowest MSE value was obtained by using the proposed CNN–CPSO algorithm, which was 1.12, representing a 60.56% decrease compared with the CNN and 37.43% decrease compared with the CNN–PSO. These significant reductions in the MSE demonstrated that CPSO possessed better hyperparameter optimization and convergence performance, and the CNN–CPSO model was the most optimal model for rainfall prediction in this study.

Figure 5 shows the comparison of the root mean square error (RMSE) of CNN, CNN–PSO and CNN–CPSO models. The conventional CNN model registered the maximum RMSE value of 1.68, while the PSO algorithm reduced it by 20.24%, leading to an RMSE of 1.34. The proposed approach, CNN–CPSO exhibited the least RMSE of 1.06, which is 36.90% less than that of the CNN model and 20.90% lower than that of the CNN–PSO model. This implies that the CPSO algorithm improves the parameter optimization process of the CNN, leading to better convergence characteristics and, ultimately, more accurate rainfall prediction models, thus making it the best-performing model for estimating rain attenuation.

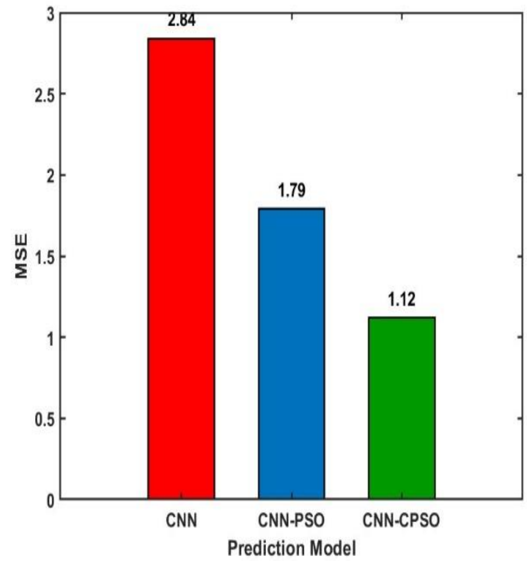

Figure 4: Comparison of Mean Squared Error (MSE)

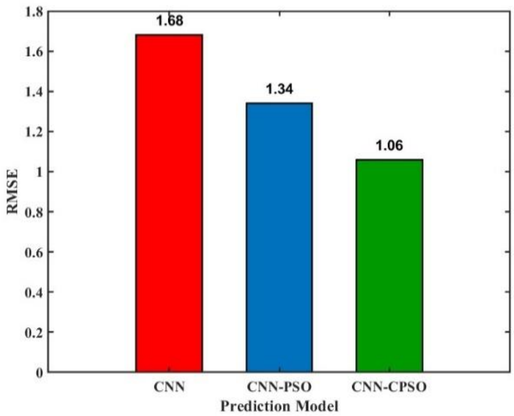

Figure 5: Comparison of Root Mean Squared Error (RMSE)

Figure 6 compares the Mean Absolute Error (MAE) of the CNN, CNN-PSO, and CNN-CPSO models. The conventional CNN achieved the highest MAE level of 1.42, whereas the error amount of CNN-PSO decreased to 1.15, which is 19.0% lower than that of the CNN. The developed CNN-CPSO model demonstrated the best performance, achieving the lowest absolute error amount of 0.89, which is 37.3% and 22.6% lower than the conventional CNN and CNN-PSO, respectively. This implies that the CNN-CPSO model can optimize CNN more accurately and stably than the existing optimization algorithms. Therefore, it is concluded that the CNN-CPSO is the most suitable model for rain attenuation prediction.

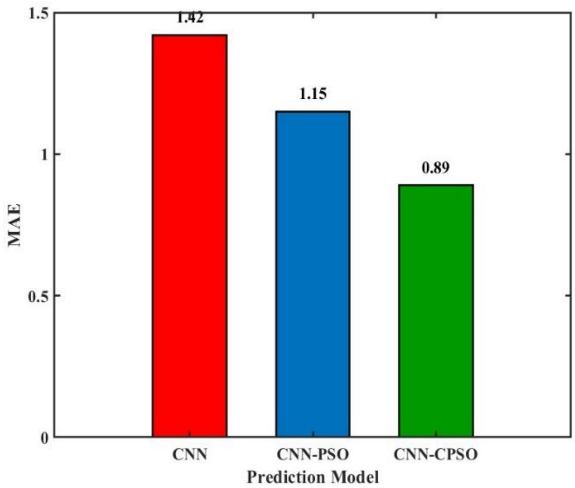

Figure 6: Comparison of Mean Absolute Error (MAE)

The Figure 7 shows the comparison results of coefficient of determination (R^2) among the CNN, CNN–PSO, and CNN–CPSO models. The coefficient of determination for the conventional CNN is 0.86, whereas for the CNN–PSO is 0.90. However, for the CNN–CPSO method, the coefficient of determination reached to 0.94, which is 9.3% higher than that of CNN and 4.4% higher than that of CNN–PSO. It can be seen that CPSO algorithm significantly improves the fitting ability and generalization ability of the CNN model, and thus, the CNN–CPSO model is proved to be the most accurate and reliable method for rainfall and rain attenuation prediction.

D. Prediction Accuracy Analysis

Figure 8 shows the predicted rainfall values versus the measured ones. The predicted data points are close to perfect regression line, which demonstrates that CNN–CPSO algorithm has a remarkable capability in modeling and predicting rainfall values. More importantly, the R^2 value reaches 0.94, which means that 94% of the variation in rainfall can be explained by the CNN–CPSO model.

To further assess the robustness of the developed model, the prediction errors were thoroughly analysed. It was found that the CNN–CPSO model had the least average prediction error. This means that the model is more stable when exposed to different levels of rainfall. Such stability is critical for accurate prediction in tropical regions, where rainfall amounts are highly variable throughout the year.

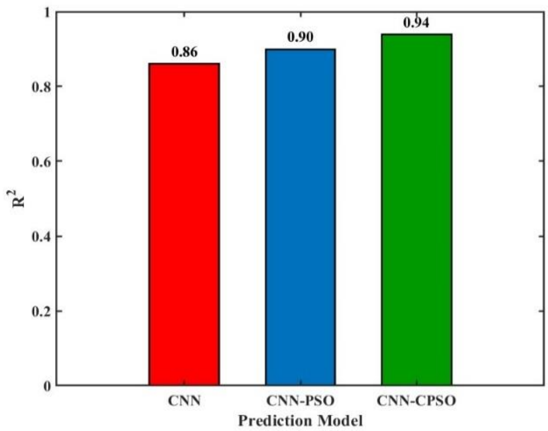

Figure 7: Comparison of Coefficient of Determination

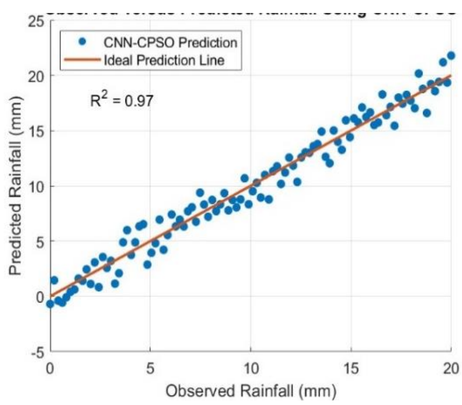

Figure 8: Observed versus Predicted Rainfall Values Using CNN–CPSO Error Distribution Analysis

Table 3: Error Analysis of Different Models

Model	Maximum Error	Average Error	Error Stability
CNN	High	1.42	Moderate
CNN–PSO	Medium	1.15	Improved
CNN–CPSO	Low	0.89	Excellent

E. Computational Efficiency Analysis

Although the algorithm contains an additional optimization phase, the achieved accuracy of the model was satisfactory without significant increase in the computational complexity, as could be seen from Table 4. Furthermore, the slight increasing of training time of the CNN–CPSO could be justified by the accuracy improvements on the prediction stage. Additionally, the optimization process allows excluding the manual search of the optimal parameter set.

Table 4: Computational Performance Comparison

Model	Training Time (min)	Optimization Process	Accuracy (%)
CNN	18.4	None	84.60
CNN–PSO	32.7	PSO tuning	87.90
CNN–CPSO	35.2	Adaptive CPSO tuning	93.30

The results from integration of the CPSO algorithm and CNN is highly efficient for the purpose of rainfall prediction. In particular, the CNN structure allows for capturing complex patterns of rainfall, while the application of CPSO enables an automatic search of optimal networks, which cannot be achieved with traditional CNNs due to their dependence on the manual selection of hyperparameters.

As shown in Figure 9, CNN-CPSO performs better than CNN-PSO which indicates the superiority of the chameleon-inspired optimization algorithm. Different to the standard PSO, CPSO has better exploration ability and has no premature convergence when used to optimize parameters. The accuracy of 93.30% and R value of 0.94 indicate that the framework could be well used to represent nonlinear behaviour of rainfall, which can be effectively used for application such as estimation in Ka-band satellite communication, rain fade mitigation and flood observation.

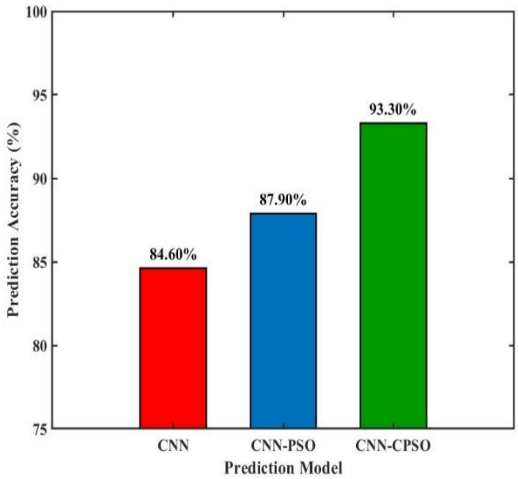

Figure 8: Rainfall Prediction Accuracy Comparison

5. Conclusion

The Enhanced CNN-CPSO framework is a significant development in intelligent rainfall prediction, combining the feature-learning ability of CNN and the adaptive optimization ability of CPSO. The optimization of CNN hyperparameters helps the model to overcome limitations of manual tuning and traditional optimization methods. The proposed architecture achieves better prediction accuracy, faster convergence and stronger generalization; thus, it is suitable for applications where reliable rainfall estimation is important. The proposed framework can be used for satellite communication, hydrology, environmental monitoring, precision agriculture, and climate modeling, in addition to rainfall forecasting, where accurate predictive analytics enable more informed choices and system resilience.

References

[1] Ajayi, G. O., & Ofoche, E. B. (1984). Some tropical rainfall rate characteristics at Ile-Ife for microwave and millimeter wave applications. Journal of Climate and Applied Meteorology, 23(4), 562–567.

[2] Shuvo, A. A. S., et al. (2026). A review of rainfall forecasting techniques: Statistical, numerical weather prediction, machine learning, and deep learning approaches. Earth-Science Reviews, 236, 104503.

[3] Semire, F. A., Mohd-Mokhtar, R., & Akanbi, I. A. (2018). Validation of new ITU-R rain attenuation prediction model over Malaysia equatorial region. MAPAN, 34(1), 37–47.

[4] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.

[6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

[7] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems, 28*, 802–810.

[8] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1948.

[9] V. Kusla et al.(2025), “Chameleon swarm algorithm with Morlet wavelet mutation for superior optimization performance,” Scientific Reports, 15,13971.

[10] Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. IEEE Access, 10, 10031–10061.

[11] Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence, 1*(1), 33–57.

[12] Javaheri, A., et al. (2022). Particle swarm optimization-based support vector regression and deep learning models for short-term rainfall forecasting. Journal of Hydrology, 611, 127981.

[13] Oliveira, T. R., et al. (2023). A comprehensive review of precipitation forecasting using machine learning and deep learning techniques. Atmospheric Research, 292, 106847.

[14] Hnin, S. W., & Jeenanunta, C. (2020). A particle swarm optimised support vector regression for short-term load forecasting. International Journal of Energy Technology and Policy, 16(4), 399–412.

[15] Umamaheswari, P. (2025). *Enhancing rainfall forecasting using a hybrid long short term–particle swarm optimization (LSTM–PSO) method*. Research Square.

[16] Pranolo, A., et al. (2023). Particle swarm optimization for deep learning hyperparameter optimization in environmental prediction. Environmental Modelling & Software, 165, 105734.

[17] Abdishakur, A. E., Tahlil, A. M., & Aden, A. A. (2025). Hybrid machine learning model for rainfall prediction using time-series data. *International Journal of Engineering Trends and Technology, 73*(6), 17–29.

[18] Satheesh, A. R., Knippertz, P., & Fink, A. H. (2025). Machine learning models for daily rainfall forecasting in northern tropical Africa using tropical wave predictors. *Weather and Forecasting, 40*(10), 1895–1916.

[19] Zhang, H., Liu, Y., Zhang, C., & Li, N. (2025). Machine learning methods for weather forecasting: A survey. *Atmosphere, 16*(1), 82.