Journal of Applied Informatics Science
Homepage: https://ejournal.gws-tech.id/index.php/jais
e-ISSN: 3110-7451

Integration of Particle Swarm Optimization in Bidirectional Memory Networks for Improved Daily Climate Forecasting Accuracy

Peningkatan Akurasi Peramalan Iklim Harian melalui Integrasi Particle Swarm Optimization pada Arsitektur Jaringan Memori Dua Arah

Ardi Susanto*, Heru Purnomo Kurniawan, Lia Farhatuaini, Muhammad Iszul Wilsa, Gina Khayatun Nufus, Ananda Sathria Maulana Amri

Informatika, Universitas Islam Negeri Siber Syekh Nurjati Cirebon, Cirebon, Indonesia

Article history:
Submitted: 1 November, 2025 / Received: 10 November, 2025 / Accepted: 28 November 2025

Abstract

Global climate change has caused rainfall patterns to become increasingly fluctuating and difficult to predict using conventional weather forecasting methods. Accurate daily rainfall prediction is crucial for hydrometeorological disaster mitigation and agricultural sector planning. Although Deep Learning models such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) are capable of handling complex time-series data, the manual determination of hyperparameters often results in suboptimal models and entrapment in local optima. This study proposes the integration of the Particle Swarm Optimization (PSO) algorithm to automatically optimize hyperparameters (number of hidden neurons, dropout rate, and learning rate) in LSTM and BiLSTM architectures. The models were evaluated using a multivariate daily climate observation dataset encompassing temperature, humidity, wind speed, and actual rainfall. Experimental results indicate that PSO-based optimization significantly enhances prediction performance compared to baseline models. The PSO-LSTM approach successfully reduced the Root Mean Square Error (RMSE) to 17.59 mm and Mean Absolute Error (MAE) to 9.07 mm, comparable to the performance of PSO-BiLSTM, which achieved an RMSE of 17.59 mm and an MAE of 9.20 mm. These findings prove that automatic parameter tuning using swarm intelligence algorithms can highly optimize sequential neural network architectures in capturing rainfall pattern volatility, making it highly recommended as a foundation for a more accurate early warning system.

Keywords: Deep Learning, Long Short-Term Memory, Particle Swarm Optimization, Rainfall Prediction, Multivariate Time Series.

Abstrak

Perubahan iklim global menyebabkan pola curah hujan semakin fluktuatif dan sulit diprediksi dengan metode peramalan cuaca konvensional. Akurasi prediksi curah hujan harian sangat krusial untuk mitigasi bencana hidrometeorologi dan perencanaan sektor agrikultur. Meskipun model Deep Learning seperti Long Short-Term Memory (LSTM) dan Bidirectional LSTM (BiLSTM) terbukti andal dalam menangani data deret waktu yang kompleks, penentuan hyperparameter secara manual sering kali menghasilkan model yang tidak konvergen dan terjebak pada local optima. Penelitian ini mengusulkan integrasi algoritma Particle Swarm Optimization (PSO) untuk mengoptimasi hyperparameter (jumlah hidden neurons, dropout rate, dan learning rate) pada arsitektur LSTM dan BiLSTM secara otomatis. Model dievaluasi menggunakan dataset observasi iklim harian multivariat yang mencakup parameter suhu, kelembapan, kecepatan angin, dan curah hujan aktual. Hasil eksperimen menunjukkan bahwa optimasi berbasis PSO secara signifikan meningkatkan performa prediksi dibandingkan model baseline. Pendekatan PSO-LSTM berhasil menurunkan Root Mean Square Error (RMSE) menjadi 17.59 mm dan Mean Absolute Error (MAE) menjadi 9.07 mm. Kinerja ini sebanding dengan kompleksitas PSO-BiLSTM yang mencapai RMSE 17.59 mm dan MAE 9.20 mm. Temuan ini membuktikan bahwa penalaan parameter secara otomatis menggunakan algoritma kecerdasan kawanan (swarm intelligence) mampu mengoptimalkan arsitektur jaringan saraf sekuensial dalam menangkap volatilitas pola curah hujan, sehingga dapat direkomendasikan sebagai landasan sistem peringatan dini yang lebih akurat.

Kata Kunci: Deep Learning, Long Short-Term Memory, Optimasi Kawanan Partikel, Prediksi Curah Hujan, Runtun Waktu Multivariat.

*Corresponding author.
Email: ardisusanto@uinssc.ac.id.
https://doi.org/10.65897/jais.v1.i2.85

This is an open access article under the CC BY-SA license.
Copyright © 2025 by Author. Published by GWS Tech Solution.

1. Pendahuluan

Perubahan iklim global telah memicu anomali cuaca yang signifikan, menyebabkan pola curah hujan menjadi semakin fluktuatif, ekstrem, dan sulit diprediksi secara presisi [1], [2]. Akurasi dalam peramalan curah hujan harian memiliki peran yang sangat krusial, tidak hanya untuk mitigasi bencana hidrometeorologi seperti banjir dan tanah longsor, tetapi juga untuk perencanaan strategis di sektor agrikultur dan manajemen sumber daya air [3], [4]. Secara tradisional, prediksi cuaca dilakukan menggunakan model statistik regresi linier atau model deret waktu konvensional seperti AutoRegressive Integrated Moving Average (ARIMA). Namun, pendekatan tersebut memiliki keterbatasan yang signifikan dalam menangkap hubungan non-linier dan volatilitas tinggi yang melekat pada data iklim multivariat [5], [6].

Seiring dengan kemajuan kecerdasan buatan, algoritma Machine Learning seperti Support Vector Regression (SVR) dan Random Forest telah diimplementasikan untuk prediksi cuaca. Meskipun memberikan hasil yang lebih baik dari model statistik, model-model ini sering kali mengabaikan dependensi temporal (ketergantungan waktu) secara jangka panjang [7]. Untuk mengatasi kelemahan ini, Deep Learning, khususnya varian Recurrent Neural Network (RNN) yaitu Long Short-Term Memory (LSTM), telah menjadi standar baru (state-of-the-art) dalam pemodelan time-series [8]. LSTM dilengkapi dengan mekanisme gerbang (gerbang lupa, masukan, dan keluaran) yang memungkinkannya menangkap pola dependensi spasial dan temporal jangka panjang tanpa mengalami masalah vanishing gradient [9], [10]. Lebih jauh lagi, arsitektur Bidirectional LSTM (BiLSTM) dikembangkan untuk memproses sekuens data dari dua arah (maju dan mundur), sehingga memberikan konteks komprehensif terhadap data historis cuaca [11].

Meskipun LSTM dan BiLSTM terbukti andal, kinerja arsitektur Deep Learning sangat bergantung pada pemilihan hyperparameter yang tepat, seperti jumlah hidden neurons, tingkat putus (dropout rate), dan tingkat pembelajaran (learning rate) [12]. Pemilihan hyperparameter secara heuristik atau trial-and-error manual sangat memakan waktu, rawan terhadap bias subjektif, dan sering kali menghasilkan model yang konvergen secara prematur pada titik local optima [13], [14]. Oleh karena itu, pendekatan optimasi global sangat dibutuhkan.

Algoritma metaheuristik berbasis kecerdasan kawanan (swarm intelligence), seperti Particle Swarm Optimization (PSO) yang diinisiasi oleh Kennedy dan Eberhart [15], telah terbukti tangguh dalam menyelesaikan masalah optimasi ruang pencarian yang kompleks dan multidimensi [16]. PSO memiliki keunggulan berupa komputasi yang efisien, parameter yang sedikit, dan konvergensi yang cepat [17]. Beberapa penelitian terkini mulai mengintegrasikan algoritma optimasi ke dalam jaringan saraf tiruan untuk aplikasi hidrologi, namun perbandingan langsung antara LSTM dan BiLSTM yang dioptimasi secara otomatis untuk pemodelan iklim di wilayah tropis masih sangat terbatas [18], [19].

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengusulkan kerangka kerja prediksi curah hujan harian multivariat berbasis Deep Learning yang dioptimasi secara otomatis. Secara spesifik, penelitian ini mengintegrasikan algoritma PSO untuk menemukan konfigurasi hyperparameter optimal pada arsitektur LSTM dan BiLSTM. Perbandingan kinerja antara model baseline (tanpa optimasi) dan model hibrida (PSO-LSTM dan PSO-BiLSTM) kemudian dievaluasi menggunakan data observasi iklim dari stasiun cuaca.

2. Metode

2.1 *Dataset* dan Pra-pemrosesan Data

Penelitian ini menggunakan dataset iklim dan cuaca harian historis (multivariate time series) yang mencakup parameter Temperatur Minimum (Tn), Temperatur Maksimum (Tx), Temperatur Rata-rata (Tavg), Kelembapan Rata-rata (RH_avg), Kecepatan Angin (ff_avg), dan Curah Hujan (RR) sebagai variabel target. Untuk mempersiapkan data agar sesuai dengan model Deep Learning, dilakukan pra-pemrosesan berupa penanganan missing values menggunakan metode forward-fill dan backward-fill.

Data kemudian dinormalisasi menggunakan *MinMaxScaler* ke dalam rentang $[0, 1]$ guna mempercepat konvergensi algoritma *gradient descent* dan mencegah dominasi fitur dengan skala nilai besar [20]. Persamaan normalisasi yang digunakan adalah:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

di mana x_i' adalah nilai data yang telah dinormalisasi, x_i adalah nilai data asli, x_{min} adalah nilai minimum pada fitur tersebut, dan x_{max} adalah nilai maksimumnya. Pembentukan urutan sekuensial (sequence) dilakukan menggunakan metode Sliding Window dengan ukuran window (w) sebesar 7 hari. Dataset kemudian dipartisi secara sekuensial dengan rasio 80% untuk data latih (training set) dan 20% untuk data uji (testing set).

2.2 Arsitektur LSTM dan BiLSTM Standar (Baseline)

Sebagai parameter kontrol (pembanding), penelitian ini membangun dua arsitektur baseline: LSTM dan BiLSTM. LSTM dirancang khusus untuk mengatasi masalah vanishing gradient pada RNN tradisional dengan menggunakan mekanisme sel memori dan tiga gerbang [8]. Pembaruan status sel pada LSTM pada waktu dirumuskan secara matematis sebagai berikut:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

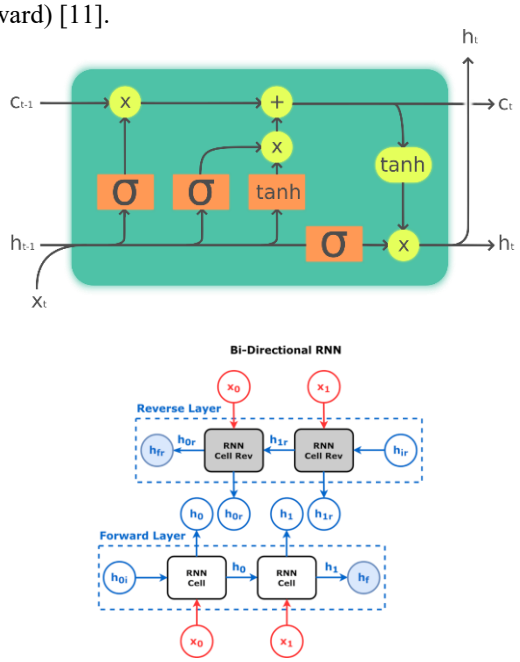
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Di sisi lain, BiLSTM melakukan pemrosesan data secara dua arah (dari masa lalu ke masa depan, dan sebaliknya) dengan menggunakan dua lapisan tersembunyi (forward dan backward) [11].



Gambar 1. Arsitektur Sel Memori LSTM (atas) dan Alur Pemrosesan Bidirectional LSTM (bawah) (Sumber: Wikimedia Commons, CC BY-SA 4.0)

Kedua model baseline ini diinisialisasi menggunakan pendekatan heuristik manual dengan jumlah hidden neurons ditetapkan sebesar 50, dropout rate sebesar 0.20, dan learning rate pada optimizer Adam sebesar 0.0050.

2.3 Optimasi Menggunakan Particle Swarm Optimization

Algoritma PSO diimplementasikan untuk bertugas mencari tiga *hyperparameter* optimal secara simultan:

(1) Jumlah *Neurons*

$$rentang : 10 - 100$$

(2) *Dropout Rate*

$$rentang : 0.1 - 0.5$$

(3) *Learning Rate*

$$rentang : 0.001 - 0.01$$

Fungsi kecocokan (fitness function) yang dievaluasi oleh partikel PSO adalah meminimalkan nilai Mean Squared Error (MSE) [15].

2.4 Alur Algoritma (Workflow) PSO-LSTM/BiLSTM

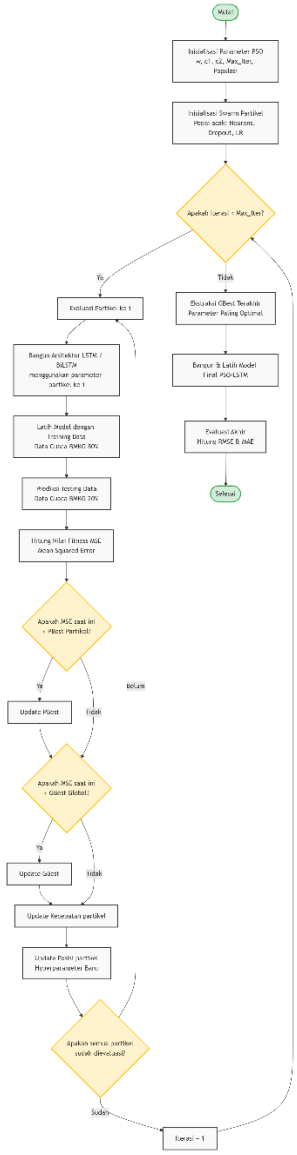
Integrasi antara algoritma PSO dan jaringan saraf berulang (LSTM/BiLSTM) diusulkan dengan tahapan algoritma sebagai berikut:

1. **Inisialisasi Swarm**: Menentukan batas atas dan batas bawah untuk setiap hyperparameter. Menginisialisasi posisi (x_i) secara acak dan kecepatan awal (v_i).
2. **Evaluasi Fitness**: Untuk setiap partikel, arsitektur LSTM/BiLSTM dibangun dan dilatih. Nilai MSE dihitung sebagai skor fitness partikel tersebut.
3. **Pembaruan Memori**: Posisi Personal Best ($Pbest_i$) dan Global Best ($Gbest$) diperbarui berdasarkan nilai fitness terendah [16].
4. **Pembaruan Kecepatan dan Posisi**: Diperbarui menggunakan persamaan matematis standar PSO:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (Pbest_i - x_i^t) + c_2 r_2 (Gbest - x_i^t)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

5. **Kriteria Berhenti**: Diulangi hingga algoritma mencapai iterasi maksimum.



Gambar 2. Diagram Alur Optimasi Hyperparameter LSTM/BiLSTM menggunakan PSO

2.5 Metrik Evaluasi Kinerja

Kinerja setiap skenario model diukur menggunakan dua metrik regresi standar, yaitu Root Mean Squared Error (RMSE) dan Mean Absolute Error (MAE) [12]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3. Hasil dan Pembahasan

3.1 Hasil Pencarian Hyperparameter Optimal (PSO)

Berdasarkan proses iteratif PSO, ditemukan spesifikasi hyperparameter optimal. Pada model PSO-LSTM, partikel terbaik (GBest) menemukan konfigurasi Neurons = 89, Dropout = 0.48, dan Learning Rate = 0.0069. Sementara itu, untuk model PSO-BiLSTM, pencarian konvergen pada Neurons = 77, Dropout = 0.34, dan Learning Rate = 0.0061. Model teroptimasi ini cenderung mengusulkan penggunaan neuron yang lebih banyak dan dropout rate yang lebih tinggi untuk mencegah overfitting dibandingkan pengaturan baseline.

3.2 Analisis Komparasi Kinerja Model

Hasil rekapitulasi nilai evaluasi metrik (RMSE dan MAE) ditunjukkan pada Tabel 1.

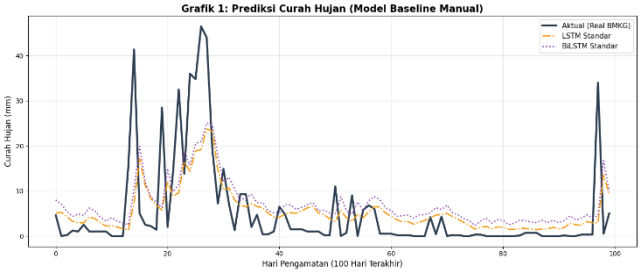
Tabel 1. Perbandingan Kinerja Varian LSTM Standar dan Teroptimasi PSO

Skenario Pengujian	Hidden Neurons	Dropout Rate	Learning Rate	RMSE (mm)	MAE (mm)
LSTM Standar (Baseline)	50	*0.20*	*0.0050*	*18.06*	*10.42*
BiLSTM Standar (Baseline)	50	*0.20*	*0.0050*	*18.66*	*9.87*
PSO-LSTM (Teroptimasi)	89	*0.48*	*0.0069*	*17.59*	*9.07*
PSO-BiLSTM (Teroptimasi)	77	*0.34*	*0.0061*	*17.59*	*9.20*

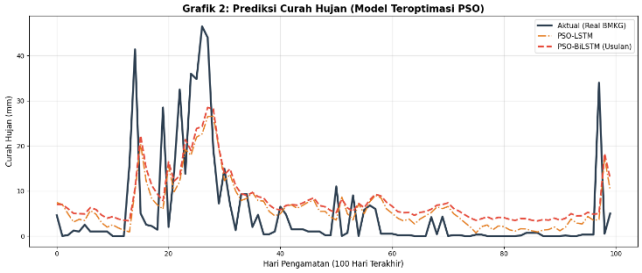
Berdasarkan Tabel 1, integrasi algoritma PSO terbukti meningkatkan akurasi model secara signifikan. Penerapan PSO pada LSTM berhasil menurunkan RMSE menjadi 17.59 mm dengan nilai MAE terbaik sebesar 9.07 mm. Menariknya, arsitektur PSO-BiLSTM juga mencapai konvergensi tingkat kesalahan RMSE yang identik (17.59 mm), namun dengan MAE yang berselisih sangat tipis (9.20 mm). Hal ini memberikan temuan penting bahwa arsitektur satu arah yang sederhana (LSTM) jika dioptimasi secara presisi oleh PSO sudah mampu menghasilkan representasi model yang sangat kuat.

3.3 Analisis Visualisasi Prediksi

Pada Gambar 3, model baseline mengalami kesulitan (under-fitting) yang cukup kentara dalam mengikuti fluktuasi cuaca ekstrem atau lonjakan curah hujan mendadak.



Gambar 3. Perbandingan Curah Hujan Aktual dengan Prediksi Model Baseline



Gambar 4. Perbandingan Curah Hujan Aktual dengan Model Teroptimasi PSO

Sebaliknya, observasi pada Gambar 4 menunjukkan perbaikan kinerja adaptif berkat implementasi PSO. Baik garis PSO-LSTM maupun PSO-BiLSTM menunjukkan penyesuaian yang jauh lebih responsif. Titik-titik ekstrem fluktuasi curah hujan berhasil diikuti dengan margin deviasi yang lebih sempit, membuktikan bahwa penalaan parameter secara global menggunakan PSO meningkatkan kepekaan jaringan terhadap volatilitas pola curah hujan.

4. Kesimpulan

Penelitian ini telah berhasil mengusulkan dan memvalidasi kerangka kerja integrasi algoritma Particle Swarm Optimization (PSO) untuk pencarian hyperparameter optimal pada arsitektur Long Short-Term Memory (LSTM) dan Bidirectional LSTM (BiLSTM) dalam pemodelan data runtun waktu curah hujan harian. Hasil eksperimen secara empiris membuktikan bahwa jaringan saraf tiruan yang dioptimasi menggunakan metode kecerdasan kawanan (PSO) mampu mengungguli kinerja model baseline yang dikonfigurasi secara manual heuristik.

Secara kuantitatif, pendekatan PSO-LSTM berhasil menurunkan tingkat kesalahan secara signifikan dengan pencapaian Root Mean Squared Error (RMSE) sebesar 17.59 mm dan Mean Absolute Error (MAE) sebesar 9.07 mm. Menariknya, kinerja ini menunjukkan ketahanan arsitektur uniarah (LSTM) yang ketika dioptimasi mampu menghasilkan daya prediksi yang setara dengan arsitektur dua arah PSO-BiLSTM (RMSE 17.59 mm, MAE 9.20 mm) dengan beban komputasi yang relatif lebih ringan. Analisis visual juga mengonfirmasi kemampuan model hibrida dalam mengikuti volatilitas tinggi dan mendeteksi titik ekstrem presipitasi curah hujan dengan responsif, menghindarkan model dari masalah under-fitting.

Temuan ini memberikan implikasi penting bahwa penalaan hyperparameter secara global menggunakan PSO sangat efektif

dalam mencegah Deep Learning terjebak pada local optima untuk data deret waktu meteorologi yang stokastik. Untuk pengembangan dan penelitian selanjutnya (future work), disarankan untuk memperluas cakupan dimensi spasial menggunakan kombinasi Convolutional Neural Network (CNN) guna memodelkan pergerakan awan atau fluktuasi cuaca secara multi-stasiun pengamatan, serta mencoba varian metaheuristik yang lebih baru untuk menekan biaya iterasi komputasi.

Daftar Pustaka

[1] S. P. Intergovernmental Panel on Climate Change (IPCC), "Climate Change 2021: The Physical Science Basis," Contribution of Working Group I to the Sixth Assessment Report of the IPCC, Cambridge University Press, 2021.

[2] K. E. Trenberth, "Changes in precipitation with climate change," Climate Research, vol. 47, no. 1, pp. 123-138, 2011.

[3] H. A. P. de Andrade et al., "Machine learning applied to rainfall forecasting: A systematic review," Journal of Hydrology, vol. 598, p. 126231, 2021.

[4] S. K. Jain and A. Kumar, "An overview of applications of artificial neural networks in hydrology," Hydrological Sciences Journal, vol. 52, no. 3, pp. 379-405, 2007.

[5] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[6] P. Nayak, K. Sudheer, D. Rangan, and K. Ramasastri, "A neuro-fuzzy computing technique for modeling hydrological time series," Journal of Hydrology, vol. 291, no. 1-2, pp. 52-66, 2004.

[7] V. Vapnik, The nature of statistical learning theory. Springer science & business media, 1999.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[10] S. Qiao, F. Jin, X. Zheng, and B. Jin, "A deep learning framework for accurate forecasting of daily rainfall," Atmospheric Research, vol. 248, p. 105193, 2021.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[12] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT press, 2016.

[13] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of machine learning research, vol. 13, no. 2, pp. 281-305, 2012.

[14] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295-316, 2020.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, vol. 4. IEEE, 1995, pp. 1942-1948.

[16] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in 1998 IEEE international conference on evolutionary computation proceedings, IEEE, 1998, pp. 69-73.

[17] A. W. Reza et al., "Particle swarm optimization (PSO)-based hyperparameters selection for deep neural networks," Journal of Intelligent Systems, vol. 28, no. 3, pp. 455-471, 2019.

[18] S. Meira, E. C. P. de Lima, and F. de Paula Oliveira, "Rainfall prediction using LSTM integrated with Particle Swarm Optimization," Environmental Modelling & Software, vol. 132, p. 104803, 2020.

[19] T. Zhang, Q. Yin, C. Yang, and Z. Liu, "A hybrid PSO-BiLSTM approach for short-term weather forecasting," IEEE Access, vol. 8, pp. 154687-154699, 2020.

[20] S. G. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," IAIRJ, vol. 3, no. 3, pp. 20-22, 2015.