Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2026, 2026:5

https://doi.org/10.28919/cmbn/9612

ISSN: 2052-2541

DECADAL RAINFALL FORECASTING USING CNN–BiLSTM: A CASE STUDY IN INDRAMAYU, INDONESIA

YUYUN HIDAYAT*, DIMAS NAUFALDY ARDIAN

Department of Statistics, Universitas Padjadjaran, Sumedang 45363, Indonesia

Abstract: Failures in rice cultivation often result from extreme rainfall variability—ranging from droughts due to low precipitation to crop damage caused by excessive rainfall. This makes rice production highly sensitive to climate fluctuations, emphasizing the need for accurate rainfall forecasting to optimize planting and harvesting schedules. In Indramayu Regency, unpredictable rainfall patterns pose significant forecasting challenges. With the Indonesian Meteorological Agency's (BMKG) average accuracy reaching only 63%, alternative approaches are urgently needed. This study employs a hybrid deep learning model—Convolutional Neural Network–Bidirectional Long Short-Term Memory (CNN-BiLSTM)—to address these limitations, as previous methods such as RNN and LSTM have struggled to capture seasonal patterns and the complexity of decadal rainfall data. The model was trained on rainfall data from January 2000 to July 2022 and tested on data from August 2022 to January 2025. The model architecture includes two convolutional layers (filters 16 and 32), max pooling, and three Bi-LSTM layers (64, 50, and 32 neurons), trained using the Adam optimizer (learning rate = 0.0001), a batch size of 64, and Mean Square Error (MSE) as the loss function. Evaluation results indicate a Mean Absolute Percentage Error (MAPE) of 17.49%, classified as "good" forecasting accuracy. This translates to an overall accuracy of 82.51% (MAPE-based classification). These findings

*Corresponding author

E-mail address: yuyun.hidayat@unpad.ac.id

Received September 23, 2025

demonstrate that the CNN-BiLSTM model effectively predicts decadal rainfall in Indramayu and has the potential to reduce crop loss by optimizing agricultural strategies such as drainage, fertilization, and harvesting aligned with rainfall projections for February to April 2025.

Keywords: forecasting; rainfall; CNN-BiLSTM; rice; Indramayu regency.

2020 AMS Subject Classification: 68T07, 92B05.

1. INTRODUCTION

West Java Province stands as one of Indonesia's key agricultural regions, often recognized as a national rice granary, with over 1.58 million hectares of rice fields [1]. In 2023, the province contributed significantly to national production, yielding 9.14 million tons of rice and ranking among the top three rice-producing provinces in Indonesia. Indramayu Regency, a major contributor to this productivity, covers approximately 125,000 hectares of paddy fields [2]. However, agriculture in Indramayu is highly dependent on rainfall patterns, which are frequently erratic and unpredictable. These fluctuations create vulnerabilities to both drought and flooding, leading to decreased agricultural yields and increased production costs.

Rice cultivation, a fundamental pillar of food security, is highly susceptible to climate variability, particularly extreme rainfall events. Both insufficient precipitation, leading to droughts due to low precipitation, and excessive rainfall, causing floods and crop damage, significantly impair agricultural productivity and farmer livelihoods. Inaccurate rainfall predictions can cause serious consequences; for instance, overestimated forecasts may delay planting schedules, resulting in drought stress and reduced yields. Between April and August 2023, droughts affected over 12,800 hectares of farmland in West Java, with 2,269 hectares specifically located in Indramayu [3]. Conversely, underestimating rainfall can lead to severe flooding, as observed in 2023 when over 113,000 hectares of agricultural land across 22 provinces were damaged [4]. Excess rainfall also contributes to soil erosion and widespread pest outbreaks [5]. Given these risks, accurate and adaptive rainfall forecasting methods are critical to improving food security in the region.

In Indonesia, rainfall is typically analyzed on a decadal scale, referring to 10-day intervals, which provides a more granular and actionable understanding of seasonal trends compared to daily or monthly aggregates. The Meteorological, Climatological, and Geophysical Agency (BMKG) utilizes this decadal classification to determine the onset of wet or dry seasons. However, the existing rainfall prediction model used by BMKG in West Java currently averages an accuracy of only 63%, exhibiting a significant error margin of 37%. This level of inaccuracy is often inadequate for effective agricultural planning and may lead to suboptimal decisions regarding crop management and planting schedules.

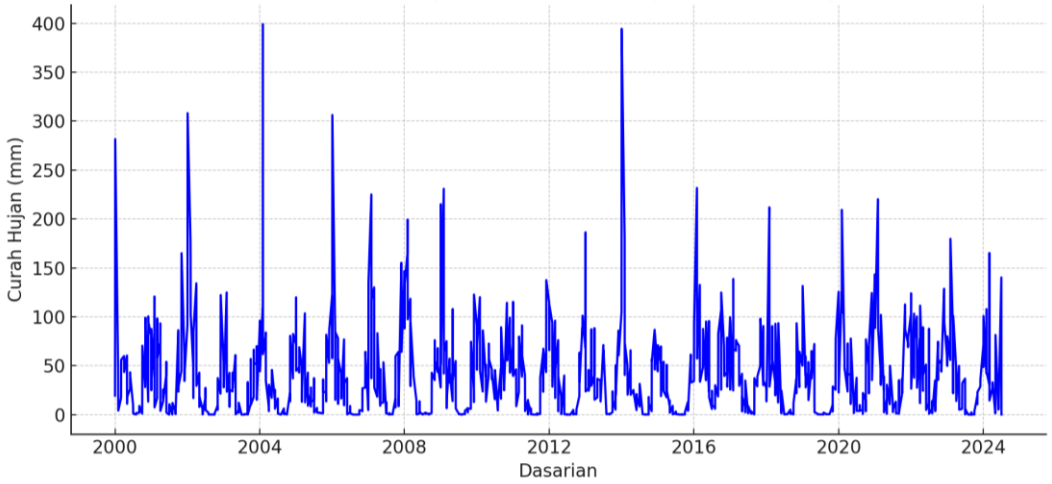

Figure 1. Rainfall Plot for Indramayu Regency, West Java Province January 2000 – July 2024

Addressing these persistent challenges necessitates more reliable and precise forecasting methods. Traditional statistical models and earlier machine learning approaches have often struggled to capture the complex, non-linear, and multi-scale temporal dependencies inherent in rainfall time series data. Recurrent Neural Networks (RNNs) and standard Long Short-Term Memory (LSTM) networks, while effective for sequential data, face limitations in discerning intricate seasonal patterns and long-term dependencies in complex decadal rainfall datasets. Studies on rainfall prediction in Bandung using deep learning have shown promising yet improvable results. For instance, an LSTM-based model achieved a test RMSE of 8.78 [6], while a BiLSTM model was evaluated with an MAE of 0.15 [7]. These findings suggest that although deep learning approaches are applicable, there remains a need for models capable of better

capturing long-term variability to support strategic agricultural planning.

Hybrid architectures such as CNN-BiLSTM offer promising solutions. CNNs are effective in extracting local features and short-term patterns, whereas BiLSTMs excel at capturing long-term temporal dependencies by processing sequences bidirectionally [8]. This integration enables simultaneous learning from both spatial and temporal representations, enhancing predictive performance. Several studies have demonstrated the superiority of CNN-BiLSTM compared to standalone models, for example in flood prediction [9] and rainfall forecasting [10].

Given these advancements and the urgent need for improved forecasting, this study aims to develop a more accurate decadal rainfall forecasting model for Indramayu Regency using a hybrid CNN-BiLSTM architecture. Specifically, this research seeks to construct a deep learning model capable of achieving at least 80% accuracy in forecasting decadal rainfall, thereby significantly outperforming current BMKG methods. This improvement is expected to support better-informed decisions regarding planting schedules, irrigation planning, and overall agricultural resource management, ultimately contributing to enhanced food security in the region.

2. MATERIALS AND METHODS

2.1. Data Source

This study employs secondary rainfall data collected from the Indonesian Meteorological, Climatological, and Geophysical Agency (BMKG). The dataset spans from the first decadal period of January 2000 to the first decadal of February 2025. It comprises 904 observations, measured in decadal intervals (10-day periods), which are standard in climatological and agricultural planning in Indonesia. The use of decadal data allows for better temporal resolution than monthly aggregates and reduces the volatility seen in daily data. The historical data used in this study is univariate time series data.

2.2. Data Preprocessing

Prior to model training, the raw rainfall data underwent several preprocessing steps to ensure optimal model performance and stability. Data normalization was performed using Min-Max scaling to rescale the rainfall values to a range between 0 and 1. This method helps to stabilize the gradient descent, prevent features with larger values from dominating the learning process, and ensure uniform input to the neural network [11]. The formula for Min-Max scaling is given by

$$x_{scaled} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

It was confirmed that the dataset contained no missing values, hence no imputation methods were required. Furthermore, no explicit handling or removal of extreme values (outliers) was performed, as the CNN-BiLSTM model's inherent robustness was relied upon, evidenced by its ability to effectively capture and predict extreme rainfall events in the results.

2.3. CNN-BiLSTM Model Architecture

The forecasting model utilized in this research is the hybrid CNN–BiLSTM model. This architecture integrates Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (BiLSTM) networks. The CNN component is responsible for extracting spatial features from the input data, such as localized patterns in decadal rainfall sequences. By applying convolutional filters over the time dimension, CNNs can detect short-term trends and local fluctuations in the signal. Mathematically, the one-dimensional convolution operation is defined as:

$$y_t = \sum_{i=0}^{k-1} x_{t+i} \cdot w_i + b \quad (2)$$

where,

y_t = output feature at time *t*

k = kernel size

x_{t+i} = input value at position $t + i$

w_i = weight of the i^{th} filter element

b = bias term

This operation allows the model to detect local temporal patterns such as abrupt spikes or drops in rainfall, which often signal the onset of extreme weather events. The kernel slides over the input sequence, producing convolved features that capture translation-invariant characteristics of the data.

Following convolution, the output is typically passed through a nonlinear activation function, such as the Rectified Linear Unit (ReLU), defined as:

$$ReLU = \max(0, x) \quad (3)$$

This function introduces nonlinearity into the model, enabling it to learn complex mappings

beyond simple linear transformations.

To further reduce computational complexity and mitigate overfitting, the CNN layer is usually followed by a pooling operation, such as max pooling, which selects the maximum value within a window:

$$p_t = \max(x_t, x_{t+1}, \cdots, x_{t+k-1}) \quad (4)$$

Pooling helps preserve the most salient features while reducing the dimensionality of the representation. In this study, the pooling size is set to 2, effectively halving the sequence length and enabling the model to focus on dominant rainfall patterns across adjacent time steps.

Once spatial features are extracted, they are passed into the BiLSTM layers, which capture temporal dependencies in the sequence. Unlike standard LSTM models, which process data in a single forward direction, BiLSTM processes the input in both forward and backward directions. This bidirectional structure allows the model to utilize contextual information from both past and future time steps, resulting in a more comprehensive understanding of temporal dynamics. The internal operations of LSTM networks are governed by gated mechanisms, including the forget gate, input gate, and output gate. The forget gate ft determines which parts of the previous cell state should be discarded:

$$f_t = \sigma(\boldsymbol{W}_f[h_{t-1}.x_t] + b_f) \quad (5)$$

Here, h_{t-1} is the hidden state from the previous time step, x_t is the current input, $\boldsymbol{W}_f$ represents the weights, and b_f is the bias. A value close to 1 means retaining the information, while a value close to 0 means forgetting it.

Next, the input gate determines what new information will be stored in the cell. This is a two-step process: the sigmoid function selects which values to update, while a tanh layer generates a vector of new candidate values to be potentially added to the cell state:

$$i_t = \sigma(\boldsymbol{W}_i[h_{t-1}, x_t] + b_i) \quad (6)$$

$$\bar{C} = \tanh(\boldsymbol{W}_c[h_{t-1}, x_t] + b_c) \quad (7)$$

These two vectors are then used to update the cell state, which is the memory component of the network:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t \quad (8)$$

This equation combines retained past information ($f_t \cdot C_{t-1}$) with new candidate memory content ($i_t \cdot \bar{C}_t$). The result is a carefully regulated internal state that allows the LSTM to maintain long-term dependences.

Subsequently, the output gate decides which parts of the updated cell state will be sent to the next hidden state. It again uses a sigmoid gate to determine importance, and multiplies the result with the tanh-activated cell state:

$$o_t = \sigma(\boldsymbol{W}_o[h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t * \tanh(C_t) \tag{10}$$

The final hidden state h_t becomes the output of the LSTM at time step t, carrying both immediate and contextual information forward.

In Bidirectional LSTM (BiLSTM) architectures, this entire process is mirrored in reverse time. A separate set of LSTM units runs backward through the sequence, and the outputs from both directions—forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$—are concatenated:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{11}$$

This dual processing allows the model to utilize both past and future context, which is particularly useful in rainfall prediction where climatic patterns can exhibit symmetrical or cyclical trends [12].

The integration of CNN and BiLSTM enables the model to simultaneously learn from spatial and temporal representations of the data, making it especially effective for modeling rainfall and climate-related phenomena. In this architecture, the input layer receives the rainfall time series data, which is then processed by the CNN layer to extract local features and short-term patterns. The extracted features are subsequently passed to the BiLSTM layer, which captures temporal dependencies in both forward and backward directions. Finally, the Dense layer generates the prediction output based on the learned representations.

2.4. Hyperparameter Tuning and Training

The training of the CNN-BiLSTM model was conducted with the Adam optimizer, a learning rate of 0.0001, a batch size of 64, and for 300 epochs. Mean Square Error (MSE) was used as the loss function during training. Hyperparameter tuning was crucial for achieving optimal model

performance. While a comprehensive grid search or Bayesian optimization was not explicitly performed due to computational constraints, an iterative empirical tuning approach was adopted. This involved evaluating various combinations of learning rates, batch sizes, and architectural configurations (e.g., number of layers, neurons, filters) based on their impact on validation loss and convergence stability. The selected values were found to yield the most consistent and favorable results, as evidenced by the stable convergence of the training and validation loss curves (Figure 2) and the improved Mean Absolute Percentage Error (MAPE) on the test set (Table 3). Regularization techniques, including Dropout layers, and L2 regularization (also known as weight decay) were applied to prevent co-adaptation, penalize large weights, and further improve generalization capacity.

2.5. Evaluation Metrics

The performance of the forecasting model was evaluated using several key metrics, including Absolute Percentage Error (APE), Mean Absolute Percentage Error (MAPE), Probability of Detection (POD), and False Alarm Ratio (FAR).

1) Absolute Percentage Error

The evaluation of the first model to be used in this research is the Absolute Percentage Error (APE). APE is used to calculate the magnitude of the error for each individual observation value. Just like MAPE, if the error value decreases, the accuracy level increases. The smaller APE values there are, the smaller the MAPE value will be because MAPE is the average of all APE values [13].

$$APE = \left|\frac{X_t - \widehat{X_t}}{X_t}\right| x100\% \quad (12)$$

where,

X_t = the observed value

$\widehat{X_t}$ = the predicted value

The smaller the APE value, the better.

2) Mean Absolute Percentage Error

The evaluation of the second model to be used in this study is the Mean Absolute Percentage Error (MAPE). MAPE is a method of measuring error in forecasting methods using the absolute error technique in each period divided by the actual observation value for that period [14]. After that, calculations are performed to obtain the average absolute percentage error. MAPE is an error

test that can measure the percentage difference between predicted data and actual data. Here is the MAPE calculation.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{X_t - \widehat{X_t}}{X_t}\right| x100\% \quad (13)$$

Where the parameter same as parameter APE.

3) Probability of Detection (POD)

POD, also known as Hit Rate, measures the proportion of observed events (actual occurrences) that were correctly predicted by the model. It is particularly useful for assessing the model's ability to detect positive instances, such as extreme rainfall events. A POD value of 1 (or 100%) indicates that all actual events were successfully predicted [15].

$$POD = \frac{Hits}{Hits + Misses} \quad (14)$$

Where Hits refer to actual extreme events that were correctly predicted as extreme, and Misses refer to actual extreme events that were not predicted as extreme.

4) False Alarm Ratio (FAR)

FAR quantifies the proportion of predicted positive events that did not actually occur. It evaluates the model's tendency to generate false alarms. A FAR value of 0 indicates no false alarms, while a value of 1 means all positive predictions were false alarms [15].

$$FAR = \frac{False\ Alarms}{Hits + False\ Alarms} \quad (15)$$

Where False Alarms refer to events predicted as extreme but which did not actually occur.

3. RESULTS AND DISCUSSION

3.1. Model Evaluation and Training Behavior

The model evaluation process began with a series of baseline experiments using simple CNN–BiLSTM configurations. These early models, featuring relatively shallow architectures with limited convolutional filters and small BiLSTM layers, failed to capture the complex temporal dynamics of decadal rainfall. As shown in Table 1, all tested configurations yielded excessively high Mean Absolute Percentage Error (MAPE) values, with test errors exceeding 140%, indicating poor predictive capability and structural underfitting.

Table 1. Baseline Parameter Combinations

Filters 1D	Kernel Size 1D	Neurons Bi-LSTM	Epoch	MAPE Train	MAPE Test
32	2	32	150	132,411	146,6516
32	2	64	150	121,291	149,403
32	2	128	150	129,518	146,0347
32	2	256	150	135,899	147,5524
32	3	32	150	137,636	141,0488
32	3	64	150	124,856	147,7695
32	2	32	150	132,411	146,6516

In response to this underperformance, the model architecture was incrementally deepened. This involved increasing the number of filters in the convolutional layers, expanding the BiLSTM layers to include more neurons, and extending the training to a full 300 epochs. However, the results—summarized in Table 2—exhibited a clear overfitting trend, where training errors dropped significantly (as low as 12.65%), while testing errors remained elevated (ranging from 44% to 65%). This divergence indicated that the model had begun memorizing the training data and failed to generalize to unseen samples.

Table 2. Parameter Combinations with Overfitting Indication

Filters 1D	Kernel Size 1D	Neurons Bi-LSTM	Epoch	MAPE Train	MAPE Test
[512,512]	7 3	[512,512,512,512]	300	23,568	65,311
[512,512,512]	9 5 5	[512,512,512]	300	21,985	45,913
[512,512,512]	9 7 4	[512,512,512]	300	18,653	44,144
[512,512,512]	7 3 3	[512,512,512,512]	300	14,477	64,676
[512,512,512,512]	9 2 7 3	[512,512,512]	300	12,649	52,211

To address the imbalance between model complexity and generalization capacity, the architecture was systematically simplified through a series of targeted refinements. The number of BiLSTM layers and their associated neurons was reduced, and convolutional layers were adjusted by reducing the number of filters to minimize the risk of overfitting. Training optimization was carried out by fine-tuning the batch size and learning rate. Most importantly, regularization techniques, including Dropout layers and L2 regularization (also known as weight decay), were applied to further improve generalization by preventing co-adaptation and penalizing large weights. Together, these design choices contributed to a more balanced architecture—one that maintains sufficient model capacity while effectively controlling overfitting and ensuring better performance

on unseen data.

This iterative process culminated in a set of configurations with consistently improved validation performance, shown in Table 3. Among them, the optimal model achieved a MAPE of 17.49% on the test set—classified as "good" forecasting accuracy [13]. The corresponding training loss and validation loss curves (Figure 2) showed consistent convergence, with both loss metrics decreasing rapidly within the first 100 epochs, followed by a gradual stabilization up to epoch 300. The minimal gap between training and validation losses throughout indicates a balanced and well-generalized model with no overfitting symptoms. These findings demonstrate that model complexity must be carefully managed to avoid under- or overfitting. The optimal architecture reflects a strategic balance between network depth, parameter count, and training dynamics, validating the CNN–BiLSTM model's suitability for high-resolution rainfall forecasting. It is important to note that the configurations presented in Tables 1, 2, and 3 represent only a subset of the total experiments conducted during model tuning; dozens of additional parameter combinations were tested and analyzed for clarity and conciseness.

Table 3. Optimal Parameter Combinations

Filters 1D	Kernel Size 1D	Neurons Bi-LSTM	Epoch	MAPE Train	MAPE Test
[32,16]	3	[64,64,50]	300	17,119	18,462
[32,16]	3	[64,50,64]	300	21,985	17,517
[32,16]	3	[64,100,64]	300	18,653	18,247
[32,16]	**3**	**[64,50,32]**	**300**	**18,140**	**17,498**
[32,16]	3	[64,100,32]	300	19,544	17,645
[32,16]	3	[64,50,50]	300	21,654	17,715
[32,16]	3	[64,50,100]	300	18,673	17,635

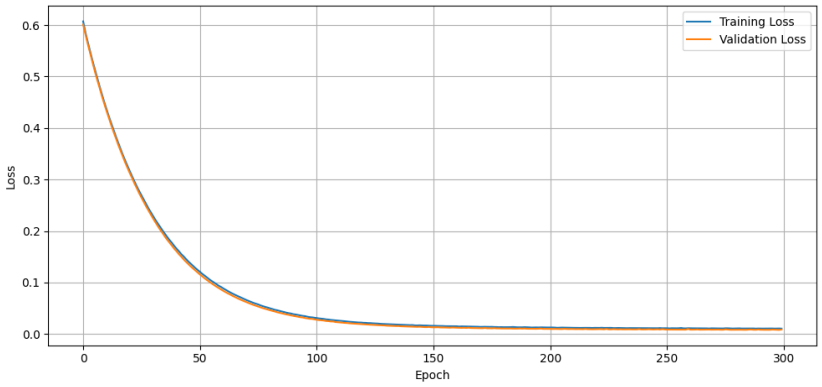

Figure 2. Training Loss and Validation Loss Curve

As depicted in Figure 3, the model is able to closely approximate the overall trend of the actual rainfall data, even though the predictions do not perfectly align with all observed values. Notably, the model demonstrates the ability to capture extreme rainfall events, supporting its robustness in handling outlier and high-variance data. Additionally, the model successfully identifies and reproduces seasonal patterns present in the actual series. These similarities can be further verified not only through the visual comparison in the graph but also through detailed numerical analysis, as shown in Table 4.

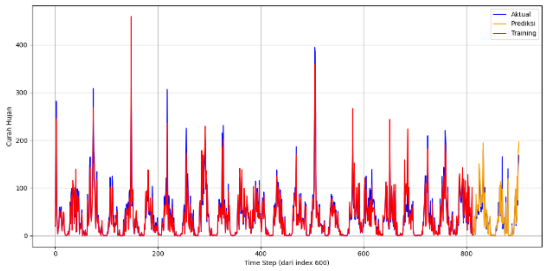

Figure 3. Comparison Between Actual and Prediction

Table 4. Comparison Between Actual and Prediction (Training)

Decadal	Actual	Prediction	APE (%)
20000101	19,38	21,28	9,839376
20000102	87,91	72,44	17,59372
20000103	281,61	244,28	13,25654
20000201	46,18	41,16	10,87826
…	…	…	…
20220603	4,78	5,50	15,15109
20220701	11,92	13,71	15,02602
20220702	87,83	101,01	15,00314
20220703	0,88	1,06	19,55512

From Table 4, it was found that 84.13% of the training data had an APE value below 20%, while the remaining 15.86% of the training data had an APE value above 20%. For the testing data (Table 5), 83.51% had an APE value below 20%, with the remaining 16.48% having an APE value above 20%. Further analysis of samples with APE values exceeding 20% indicates that these higher errors tend to occur during periods of rapid transition between seasons (e.g., onset of rainy season or transition to dry season) or during exceptionally intense rainfall events, which are inherently more challenging to predict due to their high variability and non-linear dynamics. For extreme rainfall events (e.g., >100mm), the model demonstrated a Probability of Detection (POD) of 85.37%, indicating a high success rate in identifying actual extreme events. Concurrently, the False Alarm Ratio (FAR) was found to be 26.32%, representing the proportion of predicted extreme events that did not materialize. These metrics further quantify the model's capability in identifying critical high-impact events while managing false predictions.

Table 5. Comparison Between Actual and Prediction (Testing)

Decadal	Actual	Prediction	APE (%)
20220801	6,0282	7,06	17,11622
20220802	13,9679	15,8	13,1165
20220803	1,4899	1,66	11,41687
20220901	34,9134	40,78	16,80329
…	…	…	…
20250101	106,4886	124,5915	16,99988
20250102	63,9884	75,86816	18,56548
20250103	145,5348	173,2796	19,06406
20250201	167,9847	199,5466	18,78858

Because during training and testing the model proved capable of following the rainfall pattern, and its accuracy was also quite high, the researchers have selected the model as the one to be used for future rainfall data predictions.

3.2. Prediction

After obtaining the best model, the researchers used that model to predict rainfall for the next 6 decadal (2 months ahead), specifically from February to April 2025. Here are the results of the prediction: Figure 4 and Table 6.

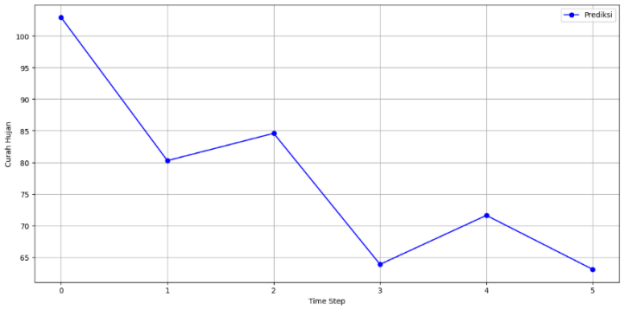

Figure 4. Prediction of Rainfall

Table 6. Prediction of Rainfall

Decadal	Prediction
20250202	102,92
20250203	80,28
20250301	84,59
20250302	63,89
20250303	71,63
20250401	63,10

As shown in Figure 4 and Table 6, the predicted rainfall values for February to April 2025 exhibit a gradual downward trend, indicating a potential transition from the peak of the rainy season toward drier conditions. While the first period shows relatively high rainfall, subsequent decadal forecasts decrease consistently, suggesting the model anticipates a shift in seasonal

dynamics. It is important to note that for this specific forecast period (February to April 2025), actual historical data is not yet available. Therefore, the rationality of this forecasted trend is verified by its consistency with typical historical rainfall patterns observed in Indramayu during these months, reinforcing the model's ability to capture overall seasonal variations. Furthermore, Figure 3 also provides a comprehensive view of how the model's predictions (orange line) align with historical actual data (blue line) across the entire dataset used for training and testing, including the test set which extends into early 2025, visually confirming the model's capability to learn and reproduce historical trends and seasonal patterns, lending confidence to its future projections.

3.3. Discussion

This study demonstrates that the CNN–BiLSTM model effectively captures temporal and seasonal patterns in decadal rainfall data, achieving a test MAPE of 17.49%. Model tuning revealed that overly simple architectures caused underfitting, while deeper ones led to overfitting, underscoring the need to balance complexity. Regularization (Dropout, L2) and careful hyperparameter tuning (learning rate, batch size) proved more effective than merely increasing depth.

The CNN–BiLSTM outperformed BMKG’s average accuracy of 63% in West Java due to its hybrid architecture. The CNN component extracted localized features and short-term patterns from decadal sequences, while BiLSTM captured long-term dependencies and cyclical trends by processing data bidirectionally. This synergy enabled a more comprehensive understanding of rainfall dynamics and improved predictive robustness.

Importantly, the model relies only on historical univariate decadal rainfall data, without requiring radar or satellite inputs, making it applicable in regions with limited infrastructure. This enhances its value for agricultural planning, particularly in data-scarce areas.

Forecast results have practical implications: projected high rainfall from February to early April 2025 suggests the need for optimized drainage, while fertilization and pesticide application should align with drier periods. The decline in rainfall toward May supports gradual field drying

before harvest. Although the model performed well in capturing extreme events (POD 85.37%, FAR 26.32%), future work should incorporate climatic variables or outlier-sensitive architectures to further enhance robustness under volatile conditions.

4. CONCLUSION

This study demonstrates the effectiveness of a hybrid Convolutional Neural Network–Bidirectional Long Short-Term Memory (CNN-BiLSTM) model in forecasting decadal rainfall in Indramayu Regency. The optimal model architecture, characterized by its combination of a 24-step input window, convolutional layers with [32, 16] filters and a kernel size of 3, L2 regularization (kernel and recurrent) at 0.001, and max pooling, effectively extracts local features and short-term patterns from the rainfall sequences. Subsequently, the Bi-LSTM layers, with [64, 50, 32] neurons, proficiently capture long-term dependencies and complex decadal cycles by processing data in both forward and backward directions. This synergistic combination allows the model to comprehensively understand and predict intricate rainfall dynamics. Training was conducted efficiently with the Adam optimizer (learning rate of 0.0001), a batch size of 64, and 300 epochs.

The CNN-BiLSTM model achieved a Mean Absolute Percentage Error (MAPE) of 17.49% on the test data, corresponding to an accuracy of 82.51%. This performance significantly outperforms the average BMKG forecast accuracy of 63% in the West Java region, affirming the model's superior ability to capture seasonal and complex rainfall dynamics. These results highlight its strong potential to enhance early warning systems and facilitate data-driven agricultural planning. Forecast results indicate that rainfall will remain relatively high from late February to early April 2025. Based on these projections, farmers are advised to optimize drainage systems and strategically schedule key agricultural activities—such as fertilization and pesticide application—during periods of lower rainfall to mitigate crop stress and maximize yield potential.

For practical implementation and to maximize its impact, future work could explore integrating this model into existing operational systems, such as those used by BMKG, potentially through an API interface. This would enable real-time and seamless deployment of the forecasts for more immediate and adaptive agricultural decision support.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Badan Pusat Statistik, Pada 2023, Luas Panen Padi Mencapai Sekitar 10,21 Juta Hektare dengan Produksi Padi Sebesar 53,98 Juta Ton Gabah Kering Giling (GKG), Badan Pusat Statistik, (2024). https://www.bps.go.id/id/pressrelease/2024/03/01/2375/pada-2023--luas-panen-padi-mencapai-sekitar-10-21-juta-hektare-dengan-produksi-padi-sebesar-53-98-juta-ton-gabah-kering-giling--gkg-.html.

[2] Badan Pusat Statistik Jawa Barat, Produksi Padi Menurut Kabupaten/Kota (Ton), 2024, (2024). https://jabar.bps.go.id/id/statistics-table/2/NTIjMg==/produksi-padi-menurut-kabupaten-kota.html.

[3] Direktorat Jenderal Tanaman Pangan, Laporan tahunan 2023, Kementerian Pertanian Republik Indonesia, (2023). https://psekp.setjen.pertanian.go.id/web/wp-content/uploads/2024/07/laptah_2023.pdf.

[4] Indonesia.go.id. Mitigasi banjir jadi sorotan utama World Water Forum. Indonesia.go.id. (2024). https://indonesia.go.id/kategori/editorial/7964/mitigasi-banjir-jadi-sorotan-utama-world-water-forum.

[5] Direktorat Jenderal Tanaman Pangan, Teknologi dan Ilmu Pengetahuan Terkait Tanaman Pangan, Kementerian Pertanian Republik Indonesia, (2024).

[6] R.F. Firdaus, Prediksi Curah Hujan Menggunakan Metode Long Short Term Memory (Studi Kasus: Kota Bandung), Thesis, Universitas Islam Indonesia, (2022).

[7] N.A. Dewanti, Prediksi curah hujan menggunakan metode Bidirectional Long Short-Term Memory (BiLSTM) di Kota Bandung, Thesis, Universitas Islam Indonesia, (2023).

[8] M. Krichen, Convolutional Neural Networks: A Survey, Computers 12 (2023), 151. https://doi.org/10.3390/computers12080151.

[9] S. Sandiwarno, Penerapan Machine Learning Untuk Prediksi Bencana Banjir, J. Sist. Inf. Bisnis 14 (2023), 62-76. https://doi.org/10.21456/vol14iss1pp62-76.

[10] R. Svs, R. Mukkamala, A CNN Bidirectional LSTM Framework for Predicting Monsoon Rainfall in India, Acta Hydrol. Slovaca 24 (2023), 214-220. https://doi.org/10.31577/ahs-2023-0024.02.0024.

[11] B. Yegnanarayana, Artificial Neural Networks for Pattern Recognition, Sadhana 19 (1994), 189-238.

https://doi.org/10.1007/bf02811896.

[12] Z. Hameed, B. Garcia-Zapirain, Sentiment Classification Using a Single-Layered BiLSTM Model, IEEE Access 8 (2020), 73992-74001. https://doi.org/10.1109/access.2020.2988550.

[13] C.D. Lewis, Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting, Butterworth-Heinemann, (1982).

[14] Mislan, A.T.R. Dani, Navigating Samarinda's Climate: A Comparative Analysis of Rainfall Forecasting Models, MethodsX 14 (2025), 103080. https://doi.org/10.1016/j.mex.2024.103080.

[15] I.T. Jolliffe, D.B. Stephenson, (Eds.) Forecast Verification: A Practitioner's Guide in Atmospheric Science, John Wiley & Sons, (2012).

[16] M. Maimaiti, A. Wumaier, K. Abiderexiti, T. Yibulayin, Bidirectional Long Short-Term Memory Network with a Conditional Random Field Layer for Uyghur Part-Of-Speech Tagging, Information 8 (2017), 157. https://doi.org/10.3390/info8040157.

[17] S. Kim, H. Kim, A New Metric of Absolute Percentage Error for Intermittent Demand Forecasts, Int. J. Forecast. 32 (2016), 669-679. https://doi.org/10.1016/j.ijforecast.2015.12.003.

[18] I.D. Mienye, T.G. Swart, G. Obaido, Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications, Information 15 (2024), 517. https://doi.org/10.3390/info15090517.

[19] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, M. Weyrich, A Survey on Long Short-Term Memory Networks for Time Series Prediction, Procedia CIRP 99 (2021), 650-655. https://doi.org/10.1016/j.procir.2021.03.088.

[20] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to Forget: Continual Prediction with LSTM, Neural Comput. 12 (2000), 2451-2471. https://doi.org/10.1162/089976600300015015.

[21] W.A. Woodward, B.P. Sadler, S. Robertson, Time Series for Data Science, Chapman and Hall/CRC, New York, 2022. https://doi.org/10.1201/9781003089070.

[22] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. Muller, Deep Learning for Time Series Classification: A Review, Data Min. Knowl. Discov. 33 (2019), 917-963. https://doi.org/10.1007/s10618-019-00619-1.

[23] L.F. Faridah, Peramalan Curah Hujan di Kabupaten Indramayu Dengan Menggunakan Metode Fungsi Transfer, Thesis, Universitas Padjadjaran. (2021). https://repository.unpad.ac.id/items/1e58b574-f6c9-4795-addd-8b49e30b68be.

[24] A.R. Fariesta, I.P.A. Shidiq, M. Dimyati, Spatial and temporal analysis of drought in rice fields using Normalized Difference Drought Index (NDDI) in Indramayu Regency, in: S. B. Wibowo, P. Wicaksono (Eds.), Proceedings of the Seventh Geoinformation Science Symposium 2021, 2021.

[25] M. Fathi, M. Haghi Kashani, S.M. Jameii, E. Mahdipour, Big Data Analytics in Weather Forecasting: A Systematic Review, Arch. Comput. Methods Eng. 29 (2021), 1247-1275. https://doi.org/10.1007/s11831-021-09616-4.

[26] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions, Int. J. Forecast. 37 (2021), 388-427. https://doi.org/10.1016/j.ijforecast.2020.06.008.

[27] J.E. Hanke, D.W. Wichern, Business Forecasting, Pearson, (2008).

[28] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, et al., State-Of-The-Art in Artificial Neural Network Applications: A Survey, Heliyon 4 (2018), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938.

[29] P.V. Hoa, N.A. Binh, P.V. Hong, N.N. An, G.T.P. Thao, et al., One-Dimensional Deep Learning Driven Geospatial Analysis for Flash Flood Susceptibility Mapping: A Case Study in North Central Vietnam, Earth Sci. Inform. 17 (2024), 4419-4440. https://doi.org/10.1007/s12145-024-01285-8.

[30] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Comput. 9 (1997), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[31] J. Kim, N. Moon, BiLSTM Model Based on Multivariate Time Series Data in Multiple Field for Forecasting Trading Area, J. Ambient. Intell. Humaniz. Comput. (2019). https://doi.org/10.1007/s12652-019-01398-9.

[32] C.M. Annur, Inilah 10 Sektor Utama Penopang Ekonomi Indonesia pada 2023, Industri Pengolahan Terbesar, (2023). https://databoks.katadata.co.id/ekonomi-makro/statistik/826961544a2f687/inilah-10-sektor-utama-penopang-ekonomi-indonesia-pada-2023-industri-pengolahan-terbesar.

[33] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward, PLOS ONE 13 (2018), e0194889. https://doi.org/10.1371/journal.pone.0194889.

[34] S.K. Morley, T.V. Brito, D.T. Welling, Measures of Model Performance Based on the Log Accuracy Ratio, Space Weather. 16 (2018), 69-88. https://doi.org/10.1002/2017sw001669.

[35] M. Kumar, S.K. Mishra, A Comprehensive Review on Nature Inspired Neural Network Based Adaptive Filter for Eliminating Noise in Medical Images, Curr. Med. Imaging Former. Curr. Med. Imaging Rev. 16 (2020), 278-287. https://doi.org/10.2174/1573405614666180801113345.

[36] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M.Z. Babai, D.K. Barrow, et al., Forecasting: Theory and Practice, Int. J. Forecast. 38 (2022), 705-871. https://doi.org/10.1016/j.ijforecast.2021.11.001.

[37] G. Zhang, B. Eddy Patuwo, M. Y. Hu, Forecasting with Artificial Neural Networks:: The State of The Art, Int. J. Forecast. 14 (1998), 35-62. https://doi.org/10.1016/s0169-2070(97)00044-7.

[38] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, M. Weyrich, A Survey on Long Short-Term Memory Networks for Time Series Prediction, Procedia CIRP 99 (2021), 650-655. https://doi.org/10.1016/j.procir.2021.03.088.

[39] T. Fischer, C. Krauss, Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions, Eur. J. Oper. Res. 270 (2018), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054.

[40] E. Surmaini, T. Hadi, K. Subagyono, N. Puspito, Prediction of Drought Impact on Rice Paddies in West Java Using Analogue Downscaling Method, Indones. J. Agric. Sci. 16(2015), 21-30.

[41] S.J. Taylor, B. Letham, Forecasting at Scale (Practical Evaluation and Comparisons Including RNN Methods), PeerJ Preprints (2018). https://doi.org/10.7287/peerj.preprints.3190v2.

[42] B. Tjasyono, Klimatologi, Penerbit ITB, Bandung, (2004).

[43] Z. Boger, D. Kogan, N. Joseph, Y. Zeiri, Improved Data Modeling Using Coupled Artificial Neural Networks, Neural Process. Lett. 51 (2019), 577-590. https://doi.org/10.1007/s11063-019-10089-7.

[44] Y. Yu, X. Si, C. Hu, J. Zhang, A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, Neural Comput. 31 (2019), 1235-1270. https://doi.org/10.1162/neco_a_01199.