CAUCHY: Jurnal Matematika Murni dan Aplikasi
Volume 10(1) (2025), Pages 389-402
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Hyperparameter Optimization Approach in GRU Model: A Case Study of Rainfall Prediction in DKI Jakarta

Fidia Raaihatul Mashfia, Suci Astutik*, Eni Sumarminingsih

Department of Statistics, Faculty of Mathematics and Natural Science, Brawijaya University Malang, Indonesia

Email: suci_sp@ub.ac.id

ABSTRACT

Rainfall is a crucial component of the hydrological system, playing a significant role in water resource management and disaster mitigation, particularly in urban areas such as Jakarta, which is prone to flooding. Reliable rainfall prediction is urgently needed to support early warning systems and enable more adaptive urban planning. This study proposed a daily rainfall prediction model based on a Gated Recurrent Unit (GRU), enhanced with feature selection using Random Forest and comprehensive hyperparameter optimization. The model development process involved several stages, ranging from data preprocessing to performance evaluation. Experimental results showed that the GRU configuration with a batch size of 64 and 128 neurons achieved the best performance, yielding an RMSE of 12.2832 and an MAE of 6.524. The model demonstrated good capability in capturing daily rainfall patterns, although it still faced limitations in predicting extreme events. This approach demonstrated the potential to improve the performance of meteorological time series-based prediction. With further testing on data from different regions or periods, the model could be further developed and utilized as part of an early warning system to support decision-making in flood risk management and short-term operational planning in the Jakarta area.

Keywords: Gated Recurrent Unit; Hyperparameter Optimization; Rainfall Prediction; Sliding Window; Time Series Forecasting

INTRODUCTION

Rainfall is a key weather element that plays an essential role in human life, particularly in water resource management, agriculture, and disaster mitigation, such as flood prevention. In urban areas like Jakarta, unpredictable rainfall patterns often lead to flooding, disrupting infrastructure and daily activities [1]. Therefore, accurate rainfall prediction is crucial to support disaster mitigation efforts and enable more effective urban planning [2]. Rainfall prediction methods have evolved significantly, ranging from conventional statistical approaches to artificial intelligence and machine learning techniques. Time series forecasting models such as Recurrent Neural Networks (RNN) and their variants have proven effective in capturing complex patterns in meteorological data. One increasingly popular model is the Gated Recurrent Unit (GRU), a variant of

Submitted: January 28, 2025 Reviewed: March 21, 2025 Accepted: May 25, 2025
DOI: https://doi.org/10.18860/cauchy.v10i1.32277

RNN designed to overcome the vanishing gradient problem while offering lower computational cost compared to Long Short-Term Memory (LSTM) networks [3], [4].

Previous studies have shown that GRU-based methods can yield more accurate rainfall predictions than conventional statistical techniques or other machine learning approaches. For instance, a study on rainfall prediction in Sidoarjo comparing 1D-CNN, RNN, LSTM, and GRU models found that GRU outperformed the others in both accuracy and computational efficiency [5]. With its strength in handling time series data, GRU emerges as a promising approach for rainfall prediction, especially in flood-prone regions such as Jakarta.

This study combines Random Forest-based feature selection with hyperparameter optimization of the GRU architecture for rainfall prediction. This approach offers advantages over prior research, which typically utilized GRU without feature selection or comprehensive parameter tuning. Random Forest was employed to select the most relevant meteorological variables, thereby improving the quality of model input. Subsequently, a thorough exploration of GRU hyperparameters, such as batch size and the number of hidden units, was conducted to identify the optimal configuration for enhancing prediction accuracy.

The contributions of this study are threefold: (1) integrating Random Forest-based feature selection with the GRU model for rainfall prediction, a method not widely applied systematically in the context of urban tropical climates; (2) applying comprehensive GRU hyperparameter tuning to improve predictive performance; and (3) testing the proposed approach specifically on daily rainfall data from Jakarta over the period 2015–2024, contributing to the development of early warning systems and data-driven disaster mitigation policies. By using this approach, the researchers aim to develop an optimized and accurate GRU-based rainfall prediction model. The resulting model is expected to be utilized as part of an early warning system and support decision-making in flood risk management and short-term operational planning in the Jakarta region.

METHODS

The methodology employed in this study consisted of several key stages, each designed to ensure the model could effectively learn rainfall patterns and produce accurate predictions. The process began with data preprocessing, followed by data splitting, feature selection using Random Forest, construction of 30-day sliding windows, GRU model training with hyperparameter optimization, and finally, model evaluation. The overall workflow of the study is illustrated in Figure 1.

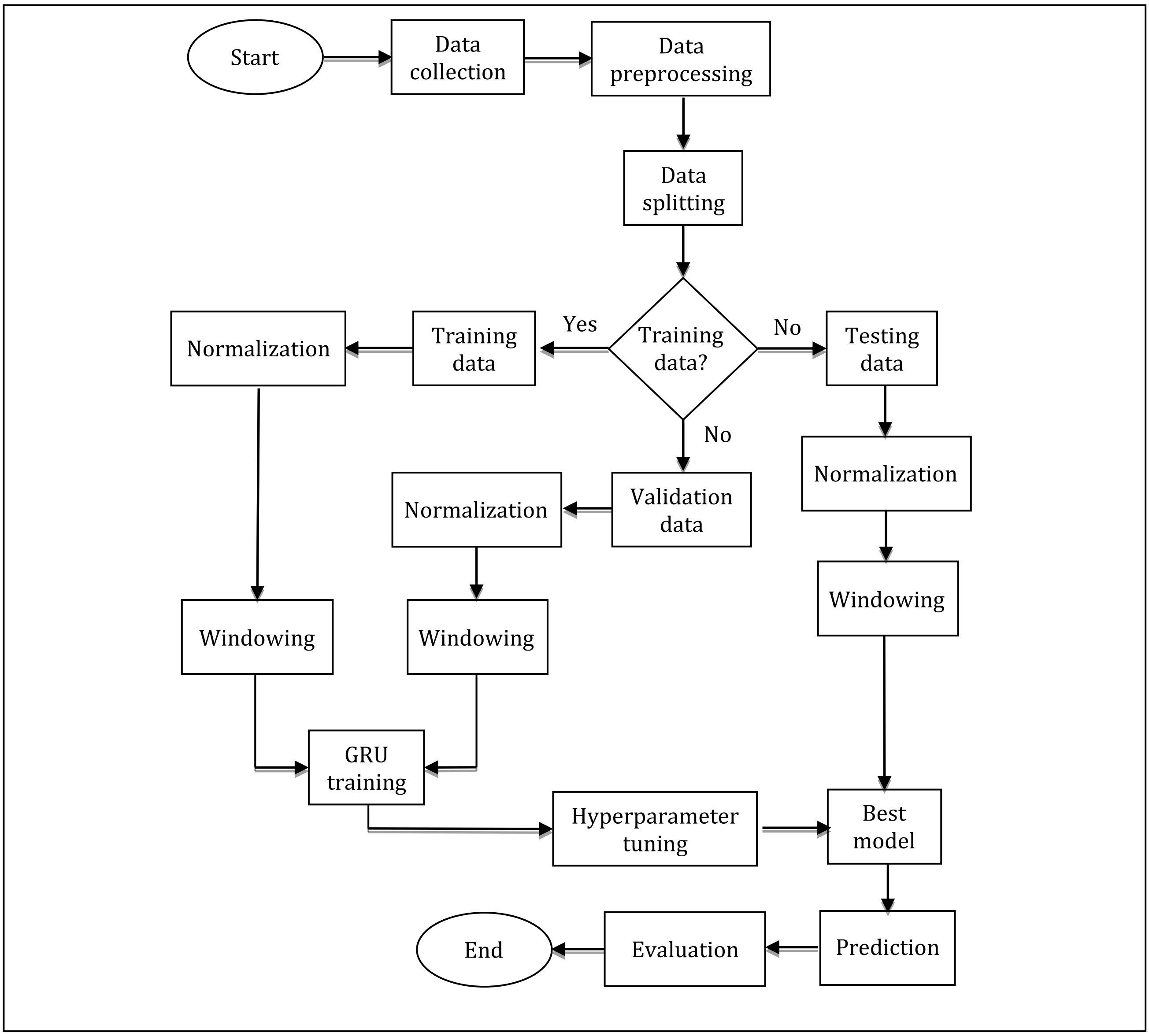

Figure 1. Flowchart of the GRU Model

Data Collection

This research utilized daily weather data from 2015 to 2024, collected at the Kemayoran Meteorological Station in Jakarta and provided by the Meteorology, Climatology, and Geophysics Agency of Indonesia (BMKG) [6]. The dataset consisted of nine weather-related features, with rainfall as the target variable and the remaining eight features serving as predictors for the modelling process. The complete list of features used in this study is summarized in Table 1 [6]–[12]:

Table 1. Features and Descriptions of Meteorological Variables

Features	Description
TN	Minimum Temperature (°C)
TX	Maximum Temperature (°C)
TAVG	Average Temperature (°C)
RH_AVG	Average Humidity (%)
SS	Sunshine Duration (jam)
DDD_X	Wind Direction at Maximum Speed (°)
FF_AVG	Average Wind Speed (m/s)
FF_X	Maximum Wind Speed (m/s)
RR	Rainfall (mm)

An example of the dataset used in this study is presented in Table 2.

Table 2. Sample Research Dataset

No.	Tanggal	TN	TX	TAVG	RH_AVG	SS	FF_X	DDD_X	RR
1	1/1/2015	24	29.4	26.1	83	0	3	270	25.1
2	2/1/2015	25	29.8	26.4	85	0	4	290	6.9
3	3/1/2015	24	29.8	26	87	0.3	5	330	31.6
...	...	...	...	...	...	...	...	...	...
3663	29/12/2024	27.8	33	29.7	73	3	3	270	0
3664	30/12/2024	24.8	32.4	28.4	76	1.8	6	240	1
3665	31/12/2024	25.2	32.4	28.4	76	2.4	5	270	1.6

Data Interpolation

The collected data underwent inspection and preprocessing to handle any missing values before further analysis. When incomplete records were found, linear interpolation was applied to fill the gaps and ensure data completeness. In this dataset, symbols such as "-" and the number "8888" were identified, which indicated missing or unavailable values in the BMKG records. To address this issue, average interpolation was employed, replacing the missing values with the mean of the preceding and succeeding values within the same column. This method was chosen to maintain data continuity without significantly altering the underlying weather patterns, allowing the model to learn more accurately [13]. Afterwards, a backfilling technique was applied to handle any remaining missing values following interpolation. Backfill is an imputation method that replaces missing values with the next available value in the data sequence [14].

Feature Selection

This study performed feature selection to identify the most influential meteorological features for rainfall prediction. A Random Forest Regressor algorithm was employed to calculate feature importance. As an ensemble learning method based on decision trees, Random Forest Regressor effectively measured the contribution of each feature to the target variable using the Mean Decrease in Impurity (MDI) method.
In regression tasks, impurity is quantified by the reduction in Mean Squared Error (MSE) at each node split. Mathematically, the contribution of a feature $\boldsymbol{i}$ in a single tree was calculated by summing the MSE reductions (ΔMSE) from each split involving that

feature, multiplied by the proportion of samples reaching the corresponding node ($P(n)$). The contribution of feature $\boldsymbol{i}$ is formulated in Equation (1) [15]:

$$\sum_{n \in N_i} P(n) \cdot \Delta MSE_n \tag{1}$$

The feature importance values were then averaged across all trees in the ensemble and normalized so that the total contribution was summed to one. Features that were frequently used and caused significant error reduction were assigned higher importance scores and considered more relevant to rainfall prediction. This method was selected due to its computational efficiency and clear interpretability regarding the relative contribution of each feature [16]. Features with very low importance, less than 10% of the total contribution, were removed to simplify the model without sacrificing accuracy. Feature selection was completed before model training to ensure only the most relevant predictors were used.

Data Splitting

To effectively train and evaluate the GRU model, the dataset was divided into three parts: 70% for training, 10% for validation, and 20% for testing. This resulted in 2,565 samples for training, 366 samples for validation, and 733 samples for testing. The training set was used to fit the model, while the validation set was employed to monitor the model's generalization ability and to implement early stopping based on the validation loss. The test set was used to assess the model's final performance. This approach was applied to reduce the risk of overfitting and to enhance the model's ability to generalize to unseen data.

Data Normalization

All input features were normalized to a range of [0,1] using Min-Max Scaling. Normalization was necessary to ensure that all features shared a consistent scale, enabling the GRU model to perform more effectively. The normalization process followed Equation (2) [17]:

$$\tilde{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Data Denormalization

Denormalization was performed to transform the predicted values back to their original scale. This step reused the previously stored *x_min* and *x_max* values for the conversion. The denormalization process followed Equation (3) [5]:

$$x_i = \tilde{x}_i(x_{max} - x_{min}) + x_{min} \tag{3}$$

where:

$\tilde{x}_i$: Normalized feature value,
x_i : original feature value,
x_{min} : minimum value of the feature in the dataset,
x_{max} : maximum value of the feature in the dataset.

By applying denormalization, the model's predictions could be interpreted using the same units as the original dataset.

Windowing

This study applied a windowing technique to transform the time series data into structured input-output pairs suitable for model training. This process used a sliding window approach, where the time window advanced one day at a time until the entire dataset was processed. This method ensured that the data became structured, allowing the model to learn historical patterns and produce more accurate predictions effectively. Each data sample consisted of 30 days of historical weather data as input and the rainfall value for the following day as the output.

Gated Recurrent Unit (GRU)

The GRU is a Recurrent Neural Network (RNN) type designed to overcome the vanishing gradient problem when processing sequential data. It consists of two main gates: the Update Gate to determine how much past information is retained and THE Reset Gate to control the influence of previous states on the current state. The mathematical equations of the GRU can be seen in Equations (4) to (7) [5]:

$$u_t = \sigma(W_u \cdot [h_{t-1},\ x_t] + b_u) \tag{4}$$

$$r_t = \sigma(W_r \cdot [h_{t-1},\ x_t] + b_r) \tag{5}$$

$$\tilde{h}_{\mathrm{t}} = \tanh(W_c \cdot [r_t * h_{t-1},\ x_t] + b_c) \tag{6}$$

$$h_t = u_t * \tilde{h}_{\mathrm{t}} + (1 - u_t)\ * h_{t-1} \tag{7}$$

di mana:

u_t : update gate,
r_t : reset gate,
$\tilde{h}_{\mathrm{t}}$: new candidate state,
h_t : updated current state,
W, b : learnable weights and biases,
σ : sigmoid activation function,
$*$: elementwise multiplication.

GRU was chosen for this study because of its simpler structure than LSTM, allowing for faster computation while maintaining high accuracy in time-series rainfall prediction [5].

Evaluation

Two main metrics are used to evaluate the model's performance in predicting rainfall: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). MAE measures the average absolute difference between actual and predicted values. The smaller the MAE, the more accurate the model is, as it indicates that the model's average error is lower. The equation for calculating the Mean Absolute Error (MAE) is provided in Equation (8) [18]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8}$$

RMSE is the square root of the average squared differences between actual and predicted values. It is more sensitive to significant errors than MAE due to squaring them. The smaller the RMSE, the better the model is at minimizing large prediction errors. The equation for calculating the RMSE is provided in Equation (8) [17]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (9)$$

Where:
y_i : actual value,
$\hat{y}_i$: predicted value,
n : number of data points.

By using both MAE and RMSE, the model evaluation becomes more comprehensive in assessing error levels and prediction quality.

RESULTS AND DISCUSSION

The dataset used in this study consisted of 3,665 daily rainfall time series records. Table 3 presents descriptive statistical analysis results to provide a general overview of the data characteristics before applying the prediction model**.**

Table 3. Descriptive Statistics of Meteorological Variables

	Mean	**Standard Deviation**	**Minimum**	**Median**	**Maximum**
Minimum Temperature (TN)	25.84	0.97	23.00	26.00	28.20
Maximum Temperature (TX)	32.38	1.43	24.80	32.60	36.80
Average Temperature (TAVG)	28.62	1.02	24.20	28.70	31.80
Average Humidity (RH_AVG)	75.97	6.16	54.00	76.00	96.00
Rainfall (RR)	6.84	16.83	0	0	277.50
Sunshine Duration (SS)	4.42	2.50	0	5.00	9.90
Maximum Wind Speed (FF_X)	4.61	1.50	0	4.00	15.00
Wind Direction at Maximum Speed (DDD_X)	252.10	99.95	0	300.00	360.00
Average Wind Speed (FF_AVG)	1.34	0.61	0	1.00	5.00

Table 3 displays the descriptive statistics of the meteorological variables utilized in this study. The average minimum temperature (TN) and maximum temperature (TX) were 25.84°C and 32.38°C, respectively, indicating a relatively stable temperature range in tropical regions. The average relative humidity (RH_AVG) was recorded at 75.97%, reflecting a consistently humid atmospheric condition. Rainfall (RR) had an average of 6.84 mm, but with a high variance (16.83) and a maximum value reaching 277.50 mm, indicating the occurrence of extreme rainfall events. Sunshine duration (SS) ranged from 0 to 9.90 hours per day, with a median of 5 hours, suggesting frequent cloudy or rainy days. These characteristics imply that Jakarta's weather tends to be humid, with high variability in rainfall and indications of extreme events that need to be anticipated in modelling efforts.

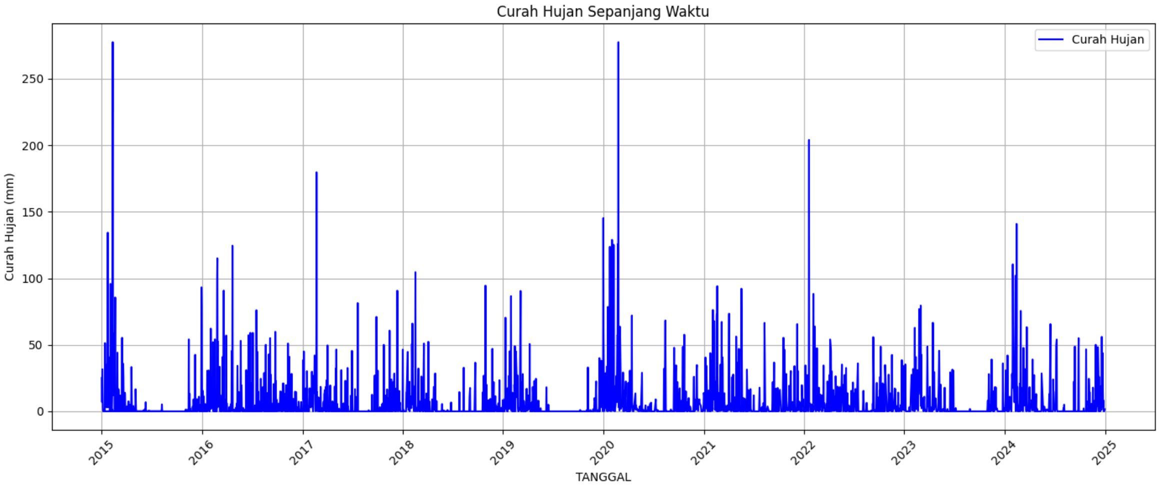

Figure 2. Rainfall from 2015 to 2024

Figure 2 illustrates the time series pattern of daily rainfall in Jakarta over the period from 2015 to 2024. The data exhibited high fluctuations, dominated by days with no rainfall or light rain, interspersed with significant spikes indicating extreme rainfall events. These spikes generally occur at the beginning and end of each year, which aligns with the seasonal rainfall pattern in tropical regions. Moreover, the relatively consistent annual pattern suggested the presence of seasonal cycles, although extreme events appeared unevenly distributed across years. These characteristics highlighted the necessity for a prediction approach capable of accurately capturing both seasonal dynamics and short-term variability.

To support this requirement, the researchers applied a preprocessing stage to ensure the completeness and quality of the input data before modelling. Linear interpolation was used as the primary method to handle missing data. When interpolation was not applicable due to consecutive missing entries at the beginning or end of the series, the backfill method was applied to maintain time series continuity. Feature selection was then performed using Random Forest to identify the most influential meteorological variables for rainfall prediction, enhancing the model input's relevance and efficiency. The results of the feature selection process are shown in Table 4.

Table 4. Feature Importance

TN	TX	TAVG	RH_AVG	SS	FF_X	DDD_X	FF_AVG
0.184	0.109	0.128	0.250	0.131	0.065	0.104	0.028

Based on Table 4, the Feature Importance results indicated that average relative humidity (RH_AVG), minimum temperature (TN), average temperature (TAVG), sunshine duration (SS), maximum temperature (TX), and wind direction (DDD_X) contributed most significantly to rainfall prediction, each with a contribution above 10%. On the other hand, features with a contribution below 10%, such as maximum wind speed (FF_X) and average wind speed (FF_AVG), were removed due to their relatively low predictive power. The GRU model was expected to operate more efficiently and improve prediction accuracy by filtering for more relevant features.

In this study, the original dataset was first divided into 70% for training, 10% for validation, and 20% for testing. Before training, all features were normalized using Min-Max Scaling to ensure uniform scale across variables and to improve training stability.

Since rainfall prediction depends on previous time series patterns, a sliding window approach was applied to structure the model input. In this approach, 30 days of historical data comprising six weather features were used as input to predict the rainfall on the 31st day as output. The choice of a 30-day window length was based on observed seasonal rainfall fluctuations (as illustrated in Figure 2), which suggested that the predictive model needed to capture short-term variability and detect extreme seasonal peaks. By using a relatively short yet informative data span, the training process aimed to enable the model to learn monthly seasonal patterns without overfitting caused by excessively long historical inputs.

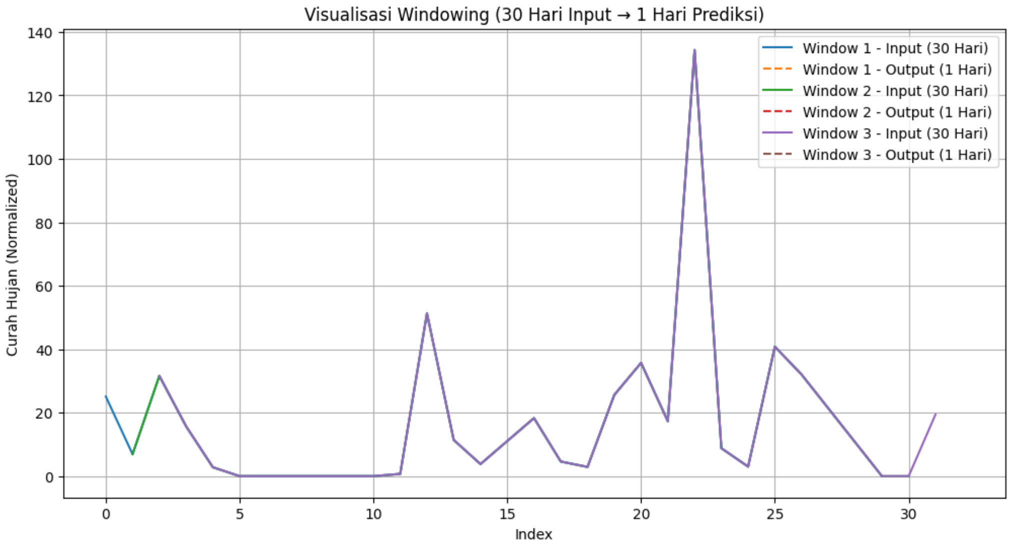

Figure 3. 30-Day Windowing Visualization for GRU Model

Figure 3 illustrates the sliding window process applied to the rainfall data, where each window consisted of 30 days of input data used to predict rainfall on the 31st day. Coloured lines represented three consecutive windows (Windows 1, 2, and 3), while the points at the end of each window denoted the target values to be predicted. The input data shifted one day at a time so that Window 2 began one day after Window 1, and so on. This visualization clarified how the sliding window technique enabled the model to learn from short-term historical patterns to improve daily rainfall prediction accuracy. The windowing process resulted in 3,636 samples, which were then divided into 2,545 training samples, 363 validation samples, and 727 testing samples. The validation set, separate from the training data, was used during model training to monitor generalization performance and to implement early stopping based on validation loss. This approach was applied consistently to both the default and tuned GRU models, ensuring that all models utilized the validation data to avoid overfitting and maintain performance on unseen data.

With the data preprocessed and structured in a sliding window format, the next step was to build a prediction model using the Gated Recurrent Unit (GRU) approach. The GRU model was selected due to its capability to capture temporal patterns in time series data such as daily rainfall. As a starting point, a baseline GRU model was implemented using PyTorch's default parameters. This model served as a performance benchmark before further parameter exploration. The full configuration of the default parameters used in the baseline model is presented in Table 5:

Table 5. Default Parameters in PyTorch

Parameters	Values
Hidden Layer	64
Layer	1
Dropout rate	0.0
Optimizer	Adam
Learning rate	0.001
Batch size	64
Loss function	MSE
Epoch	50

Table 6. Results for Default Parameters

RMSE	MAE	Train Loss	Val Loss
12.5792	6.4244	0.0023	0.0021

The evaluation results of the baseline GRU model are presented in Table 6. The model demonstrated reasonably good performance, achieving an RMSE of 12.5792 and an MAE of 6.4244. Throughout the 50 training epochs, both training and validation loss exhibited stable trends without significant fluctuations, with final values of 0.0023 and 0.0021, respectively. This consistency and balance indicated that the model did not suffer from overfitting and was able to generalize well to the validation data. These results showed that even with a default configuration, the baseline GRU model was capable of effectively capturing daily rainfall patterns, yielding relatively low RMSE and MAE values, and thus served as a reference point for comparing performance with tuned models.

Although the baseline GRU model showed promising performance, further exploration through hyperparameter tuning was necessary to identify the most optimal model configuration. This process involved varying the number of neurons in the hidden layer and adjusting the batch size, with the goal of incrementally improving predictive accuracy and testing the model's consistency across different configurations. The tuning process employed a single hidden layer, a dropout rate of 0.2, a learning rate of 0.005, the Adam optimizer, MSE as the loss function, and a maximum of 50 training epochs. The results of the tuning experiments are summarized in Table 7.

Table 7. Hyperparameter Tuning Results for GRU Model

Batch Size	Hidden Neuron	Epoch	RMSE	MAE	Loss	Val Loss
32	32	18	12.4798	7.0412	0.0023	0.0020
	64	16	12.845	6.8996	0.0023	0.0021
	128	24	12.5298	7.3178	0.0022	0.0020
64	32	26	12.8982	6.3286	0.0023	0.0022
	64	26	13.5145	7.7608	0.0022	0.0024
	128	17	12.2832	6.524	0.0023	0.0020

The evaluation of the GRU hyperparameter tuning results is shown in Table 7, which displays the combinations of batch sizes and the number of neurons in the first hidden layer. According to the results, the lowest validation loss of 0.0020 was achieved by three configurations: batch size 32 with 32 neurons, batch size 32 with 128 neurons, and batch size 64 with 128 neurons. Among these, the configuration with batch sizes 64 and 128 neurons demonstrated the best performance, with an RMSE of 12.2832 and an MAE of 6.524, the lowest values across all experiments. This indicated that although the performance differences among configurations were not substantial, hyperparameter tuning could improve predictive performance, particularly by reducing the MAE, which reflects the average absolute error. Therefore, the configuration with batch sizes 64 and 128 neurons was selected as the best-performing model, as it produced more accurate and stable daily rainfall predictions compared to other configurations.

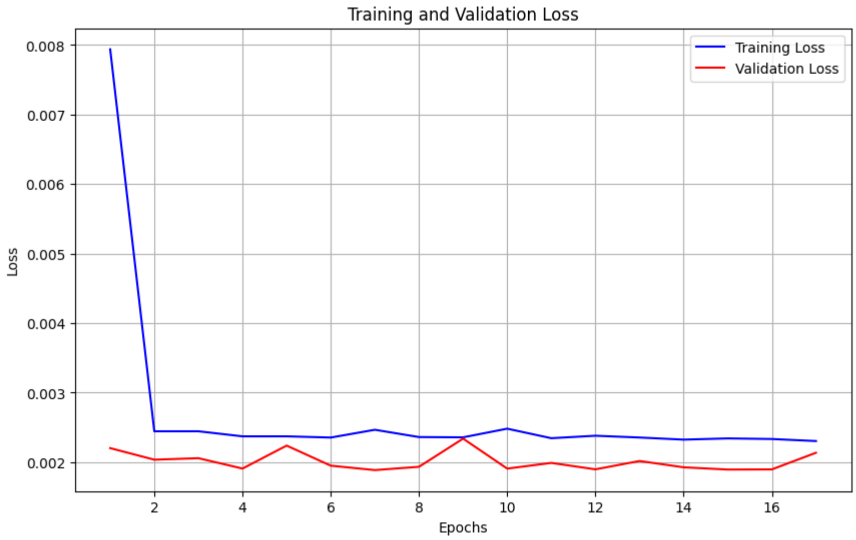

Figure 4. Training and Validation Loss Curve (Batch Size: 32, Neurons: 128)

Figure 4 shows the training and validation loss curves for the best configuration: GRU with a batch size of 64 and 128 neurons. In the early training phase, there was a sharp decline in loss, with training loss dropping drastically from approximately 0.008 to below 0.003 within the first two epochs. Subsequently, the loss curve stabilized with minor fluctuations and did not exhibit any increasing trends that would indicate overfitting. The validation loss also maintained consistent values and was relatively lower than the training loss, ranging between 0.0018 and 0.0023. This pattern indicated that the model not only learned effectively from the training data but also generalized well to the validation set. The stability of both curves reinforced that this configuration provided optimal and balanced performance during the GRU model training process.

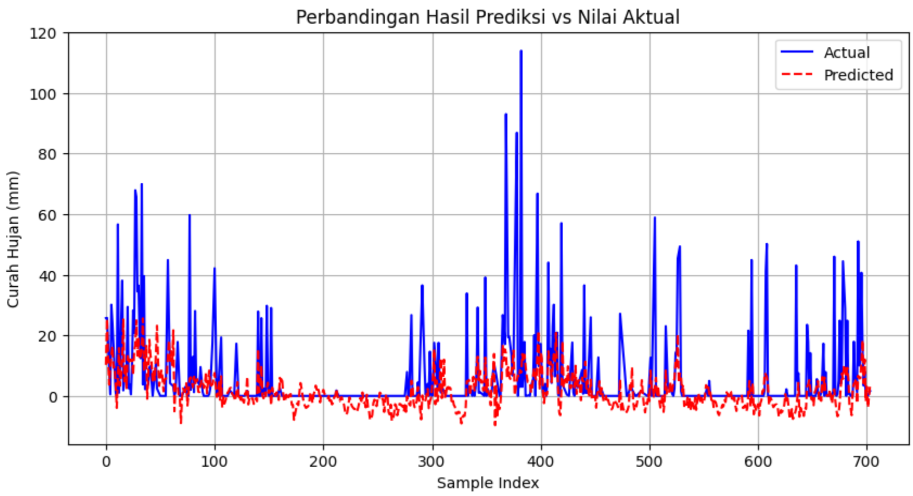

Figure 5. Plot of Predicted vs Actual Results

Figure 5 presents a comparison between actual and predicted rainfall values using the GRU model with a configuration of batch size 32 and 128 hidden neurons. In general, the model exhibited a good ability to follow the basic trend of daily rainfall, especially during dry periods or low-intensity rainfall. This was reflected in predictions approaching zero when no rainfall occurred and relatively aligned fluctuations with the actual pattern. However, during extreme rainfall events (actual values > 50 mm), the model struggled significantly to capture the sudden spikes in intensity. The predictions tended to underestimate the peak values, resulting in notable deviations.

Further analysis of these extreme events revealed that, out of a total of 15 occurrences in the test data, the model yielded an MAE of 54.74 mm and an RMSE of 57.37 mm, indicating a limitation in the model's responsiveness to high rainfall events. Despite the hyperparameter tuning, performance in extreme cases did not show adequate improvement. Therefore, further enhancements such as architectural modifications, integration of external weather data, or the application of specialized methods to handle data imbalance could be considered. Overall, the GRU model in this study successfully captured daily rainfall patterns reasonably well but still required improvements in predicting extreme events, which are critical in hydrological applications and disaster mitigation efforts.

As additional context for the performance of the developed model, Table 8 presents a numerical comparison with several prior studies that employed various approaches for daily rainfall prediction in Indonesian regions with climatic characteristics like those of Jakarta, such as Palembang, Sidoarjo, and Lampung. It is important to note that although these locations share a tropical monsoon climate comparable to Jakarta, differences existed in data periods, data sources, and feature sets used in each study. Therefore, this comparison is indicative in nature and aimed at providing a general perspective on the model’s relative performance within the relevant literature.

Table 8. Performance Metrics of Rainfall Prediction Models Across Studies

Method	Location	RMSE	MAE	R-square
GRU (This Study)	Jakarta	12.28	6.52	-
GRU [19]	Palembang	9.33	-	0.54
LSTM [19]	Palembang	7.45	-	0.70
LSTM [20]	Lampung	16.81	-	-
GRU [5]	Sidoarjo	-	-	0.79

Table 8 shows that the performance of the GRU model in this study remained competitive compared to several previous studies. The model achieved an RMSE of 12.28 and an MAE of 6.52, which, although higher than the results reported in [19] for GRU and LSTM models (with RMSEs of 9.33 and 7.45, respectively), still indicated a reasonably robust and reliable performance particularly considering that no external features were used. Another study [20] even reported lower performance, with an LSTM RMSE of 16.81, highlighting that prediction quality heavily depends on model configuration and local data characteristics. Meanwhile, the GRU model in [5] reported an R^2 of 0.79, indicating good explanatory power, although RMSE and MAE values were not provided.

Overall, the tuned GRU model in this study was effective in capturing daily rainfall patterns. Differences in performance across studies emphasized the importance of input feature selection, data preprocessing, and hyperparameter tuning. Although not the top-performing model, its results demonstrated strong potential and warranted further development, especially through the integration of climate variables or hybrid modelling approaches to enhance performance in predicting extreme rainfall events.

CONCLUSIONS

This study successfully developed a daily rainfall prediction model based on GRU, incorporating feature selection using Random Forest and comprehensive hyperparameter optimization. The baseline GRU model delivered reasonably good initial performance, and further tuning produced an optimal configuration with a batch size of 64 and 128 neurons, yielding an RMSE of 12.2832 and an MAE of 6.524. These results indicated that hyperparameter tuning consistently improved the model's predictive performance, even though the gains were not drastic. The model proved effective in identifying general daily rainfall patterns but remained limited in accurately predicting high-intensity extreme events.

For future research, it is recommended to explore more complex model architectures or implement hybrid approaches such as integrating GRU with attention mechanisms or convolutional neural networks (CNNs) and enhancing the model's ability to detect extreme rainfall patterns. Bayesian-based approaches may also be considered, either by developing Bayesian GRU architectures to account for predictive uncertainty or by applying Bayesian Optimization in the hyperparameter tuning process to achieve more efficient and adaptive results. Additionally, the incorporation of external supporting data, such as climate indices (e.g., ENSO and MJO) or spatial information from satellite imagery, could enrich the model inputs. Evaluating model performance by season and conducting targeted testing on extreme rainfall events are also essential to ensure the model is not only accurate in general but also reliable for disaster risk mitigation contexts. Furthermore, it should be noted that evaluation metrics such as MAE and RMSE, while commonly used, tend to be less sensitive to large errors associated with extreme rainfall. Therefore, future research may consider employing alternative loss functions such as Huber loss or quantile loss to improve predictive performance, particularly in forecasting high-intensity rainfall events.

REFERENCES

[1] BNPB, "Dampak Banjir di Jakarta Tahun 2021," Laporan Resmi BNPB, 2021.

[2] BMKG, "Analisis Perubahan Iklim dan Pengaruhnya terhadap Curah Hujan di Jakarta," Jurnal Meteorologi Indonesia, vol. 9, no. 2, 2022.

[3] A. Sharma and R. K. Aggarwal, "A Survey of Rainfall Prediction Using Deep Learning," in Proceedings of the IEEE International Conference on Artificial Intelligence and Machine Learning (AIML), 2021, pp. 1–6. doi: 10.1109/AIML.2021.9664730.

[4] Y. Liu et al., "Gated Recurrent Units for Time-Series Weather Prediction," IEEE Transactions on Neural Networks, vol. 32, no. 6, pp. 1223–1234, 2020.

[5] H. F. Nurrohman, R. Rochimah, D. C. R. Novitasari, A. Taufiq, F. Setiawan, and A. Hamid, "Rainfall Prediction Using Gated Recurrent Unit Based on DMI and Nino3.4 Index," 2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2022, pp. 191–196, doi: 10.1109/IAICT55358.2022.9887474.

[6] BMKG, "Data Curah Hujan Jakarta 2015-2024," Pusat Data BMKG, 2024.

[7] R. Tan et al., "Effect of Temperature on Rainfall Patterns," Climate Journal, vol. 45, no. 3, 2019.

[8] K. Prasetyo, "Time Series Analysis for Rainfall Prediction in Jakarta," Jurnal Ilmu Cuaca, vol. 12, no. 1, 2020.

[9] BMKG, "Humidity and its Correlation with Rainfall in Indonesia," Indonesian Climate Research, vol. 18, 2021.

[10] M. Zhang et al., "Solar Radiation and Precipitation Interaction: A Deep Learning Approach," Environmental Science Journal, vol. 28, no. 4, 2022.

[11] L. Wang et al., "Wind Direction and Rainfall Distribution: A Case Study in Tropical Regions," Journal of Meteorological Studies, vol. 39, no. 2, 2019.

[12] R. I. Rakhmalia, A. M. Soleh, and B. Sartono, "Pendugaan Curah Hujan dengan Teknik Statistical Downscaling Menggunakan Clusterwise Regression Sebaran Tweedie," Jurnal XYZ, vol. 4, no. 3, pp. xx–xx, 2020. DOI: 10.29244/ijsa.v4i3.667.

[13] M. R. Ismail, "Perhitungan Data Curah Hujan yang Hilang dengan Menggunakan Metode Interpolasi Linear," Jurnal Ilmiah Mahasiswa Teknik Sipil, vol. 4, no. 02, 2023. doi: 10.33365/sendi.v4i02.5023.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[15] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, “Understanding variable importances in forests of randomized trees,” in Advances in Neural Information Processing Systems, vol. 26, 2013.

[16] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[17] S. Wijaya, T. B. Kurniawan, E. S. Negara, and Y. N. Kunang, “Rainfall Prediction in Palembang City Using the GRU and LSTM Methods,” Journal of Data Science, vol. 2023, no. 04, pp. 1–10, Mar. 2023, doi: 10.61453/jods.v2023no04.

[18] J. R. Ban, Q. Gou, and Y. S. Li, "Study on Rainfall Prediction of Yibin City Based on GRU and XGBoost," *Proc. 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, 2022, pp. 1-6, doi: [1]

[19] Wijaya, S., Kurniawan, T. B., Negara, E. S., & Kunang, Y. N. (2023). *Rainfall Prediction in Palembang City Using the GRU and LSTM Methods*. *Journal of Data Science*, 2023(04), eISSN 2805-5160. Available online: https://doi.org/10.xxxx/yyyy (Accessed: May 3, 2025).

[20] Carnegie, M. D. A., & Chairani. (2023). *Perbandingan Long Short Term Memory (LSTM) dan Gated Recurrent Unit (GRU) Untuk Memprediksi Curah Hujan*. *Jurnal Media Informatika Budidarma*, 7(3), 1022-1032. ISSN 2614-5278. DOI: 10.30865/mib.v7i3.6213.