

Performance Comparison of LSTM and Transformer Model for Rainfall Data Imputation on Automatic Weather Station

Performance Comparison of LSTM and Transformer Model for Rainfall Data Imputation on Automatic Weather Station

Tri Nurmayati^{a,1}, Agung Budi Susanto^{a,2}, Arya Adhyaksa Waskita^{a,3}

^{a,b,c} Master of Informatics Engineering Program, Universitas Pamulang, South Tangerang, 15417, Indonesia
trinurmayanti@gmail.com, dosen02680@unpam.ac.id, aawaskita@unpam.ac.id

ABSTRACT

High-resolution rainfall observations from Automatic Weather Stations (AWS) are essential for hydrometeorological analysis, yet operational sensor and communication failures frequently create missing values. This study compared Long Short-Term Memory (LSTM) and Transformer architectures for imputing missing 10-minute rainfall observations recorded at the Banten Climatological Station AWS in South Tangerang during 2021-2025. Rainfall was modeled using a sliding window containing rainfall at the previous three intervals together with concurrent air temperature and relative humidity. The chronologically ordered dataset comprised 242,497 valid samples and was divided into training, validation, and testing subsets using a 60:20:20 ratio. Each architecture was trained with the Adam optimizer for up to 100 epochs under batch sizes of 32, 16, and 8, with early stopping used to control overfitting. Performance was assessed using root mean square error, mean absolute error, and the coefficient of determination. The best LSTM configuration used a batch size of 8 and achieved an RMSE of 1.1520, an MAE of 0.5140, and an R^2 of 0.6626 on the test set. The Transformer with a batch size of 8 obtained an RMSE of 0.8287, an MAE of 0.3635, and an R^2 of 0.8244. These results demonstrate a tangible success of the Transformer model, which reduced the RMSE by 28.1% and MAE by 29.3% compared to the LSTM, alongside an absolute R^2 improvement of 0.1618. This proves that self-attention represented short-range multivariate dependencies more effectively than recurrent memory under the imbalanced rainfall distribution.

Keywords : automatic weather station; imputation; rainfall; LSTM; transformer..

Submitted:

Article Info :
Reviewed:

Accepted :

Copyright © 2026 – CSRID Journal. All rights reserved.

1. INTRODUCTION

Automatic Weather Stations (AWS) provide high-frequency observations that support short-duration rainfall analysis, urban flood monitoring, water-resource management, and climate services. The value of these applications depends not only on temporal resolution but also on continuity. Sensor malfunction, power interruption, data-logger failure, communication disruption, and maintenance can create missing intervals that interrupt the rainfall sequence and weaken downstream hydrometeorological analysis [1]-[3]. This problem is especially relevant in tropical urban environments, where convective rainfall can intensify within minutes and where a single missing 10-minute record may conceal an important stage of an event.

The AWS examined in this study is operated at the Banten Climatological Station in South Tangerang, Indonesia. It records rainfall, air temperature, and relative humidity at 10-minute intervals. Rainfall is more difficult to reconstruct than comparatively smooth meteorological variables because it is intermittent, nonnegative, highly right-skewed, and dominated by dry observations. The same series contains long periods near zero and a small number of intense values, so an imputation model must avoid false rainfall during dry periods while remaining responsive to abrupt wet-period changes. These characteristics make simple deletion, mean substitution, or linear interpolation potentially unsuitable because they may discard information, reduce variance, or smooth physically sharp transitions [5]-[8].

Deep-learning models offer an alternative because they can learn nonlinear relationships from multivariate temporal inputs. Long Short-Term Memory (LSTM) networks process observations recurrently and use gated memory to preserve relevant historical information, thereby reducing the vanishing-gradient limitation of conventional recurrent networks [9]. This architecture has been applied to hydrological and meteorological series in which current conditions depend on preceding observations [10], [11]. In rainfall imputation, an LSTM can combine recent rainfall lags with temperature and humidity to estimate a missing target, but its effectiveness depends on sequence length, optimization settings, and whether the available history contains enough temporal structure.

Transformer models replace recurrence with self-attention, enabling the model to assign different relevance weights to input positions and variables [12]. Attention-based architectures have achieved strong results in multivariate time-series representation learning, forecasting, and imputation because they can model interactions without compressing the entire sequence into a single recurrent state [13]-[15]. Studies on incomplete time series also indicate that Transformer-based approaches can outperform recurrent models in some missing-data settings, although the outcome remains dependent on data scale, masking pattern, architecture, and the temporal characteristics of the target [16], [17].

Existing reviews confirm that no single imputation method is universally superior. Performance changes with the missingness mechanism, the proportion and duration of gaps, variable distribution, sampling interval, and evaluation protocol [6], [18], [19]. Previous state-of-the-art studies have demonstrated the efficacy of Deep Learning models, such as LSTM and Transformer, for multivariate time-series imputation [13]-[15]. However, a distinct research gap remains: meteorological imputation studies have frequently relied on daily, monthly, or spatially distributed observations in subtropical climates, which exhibit relatively stable patterns. Operational AWS archives in tropical regions, conversely, contain hundreds of thousands of short-interval (10-minute) records from a single station that are highly right-skewed and heavily dominated by zero values (dry periods). A model that performs well on smoother daily series may fail to capture the abrupt convective changes in this high-resolution tropical data.

A further methodological issue is that model selection should not rely on one error measure. Root Mean Square Error (RMSE) emphasizes large deviations, Mean Absolute Error (MAE) represents average absolute error, and the coefficient of determination (R^2) measures how much observed variation is represented by the estimates. These metrics can rank configurations differently, particularly for imbalanced rainfall data; therefore, their joint interpretation is required to distinguish low average error from adequate representation of rainfall variability [17], [20].

This study compares LSTM and Transformer models for reconstructing missing 10-minute rainfall observations from the South Tangerang AWS during 2021-2025. Each sample uses rainfall from the preceding 10, 20, and 30 minutes together with air temperature and relative humidity. The dataset is divided chronologically into training, validation, and testing subsets so that model evaluation reflects prediction on later observations rather than random mixing of past and future records.

To bridge this gap, the novelty of this study lies in providing a direct, apple-to-apple comparison between a recurrent architecture (LSTM) and a self-attention mechanism (Transformer) explicitly applied to handle the extreme imbalance of 10-minute tropical rainfall data. The scientific contribution is an architecture-focused assessment under an identical preprocessing pipeline, temporal input structure, and evaluation framework. By isolating the difference between recurrent memory and self-attention, the analysis identifies how batch size influences each model, explains the relationship between convergence and test performance, and provides operational considerations for using the selected model in AWS quality-control workflows.

2. METHOD

A. Study Area and Observation Site

The study used secondary observations obtained from the AWS Center of the Indonesian Agency for Meteorology, Climatology, and Geophysics. The station is located at the Banten Climatological Station in South Tangerang, and the observation period spans 2021-2025. The selected variables were rainfall (RR, mm per 10 minutes), air temperature (T, °C), and relative humidity (RH, %). Rainfall was the imputation target, whereas temperature, humidity, and three recent rainfall values were predictors. Table 1 summarizes the source, temporal resolution, variables, and number of observations used to build the model-ready dataset.

Table 1. Characteristics of AWS Dataset

Characteristis	Value
Observation site	Banten Climatological Station AWS, South Tangerang
Period	2021-2025
Temporal resolution	10 minutes
Initial observations	251,077
Model-ready samples	242,497
Target	RR(t), rainfall at the current interval
Predictors	RR(t-1), RR(t-2), RR(t-3), T(t), RH(t)

As shown in Table 1, 251,077 raw observations were available, while 242,497 samples remained after temporal feature construction and interval-consistency screening. The difference does not indicate arbitrary deletion; it mainly reflects the requirement that every target have complete lag values at 10, 20, and 30 minutes. This distinction is important because only samples with a valid temporal context can be used as ground truth for controlled missing-value reconstruction.

B. Quality Control and Missing Value Simulation

Timestamps were converted to a consistent datetime format, sorted chronologically, and examined for duplicates, irregular intervals, and structurally incomplete records. Physical-range screening was applied to prevent implausible meteorological values from becoming training targets. The observed rainfall value at each eligible time step was treated as ground truth and hidden from the model during the reconstruction task, while its historical rainfall lags, temperature, and relative humidity remained available. This supervised masking procedure made it possible to compare imputed values with known observations under the same data conditions for both architectures.

C. Sliding Window Representation

A sliding window converted the continuous observations into supervised input-target pairs. For a target at time t , the input vector contained $RR(t-1)$, $RR(t-2)$, and $RR(t-3)$, corresponding to rainfall 10, 20, and 30 minutes earlier, together with $T(t)$ and $RH(t)$. The output was $RR(t)$, which represented the value to be reconstructed. Figure 1 illustrates how the same window moves forward by one 10-minute interval while retaining overlapping historical information.

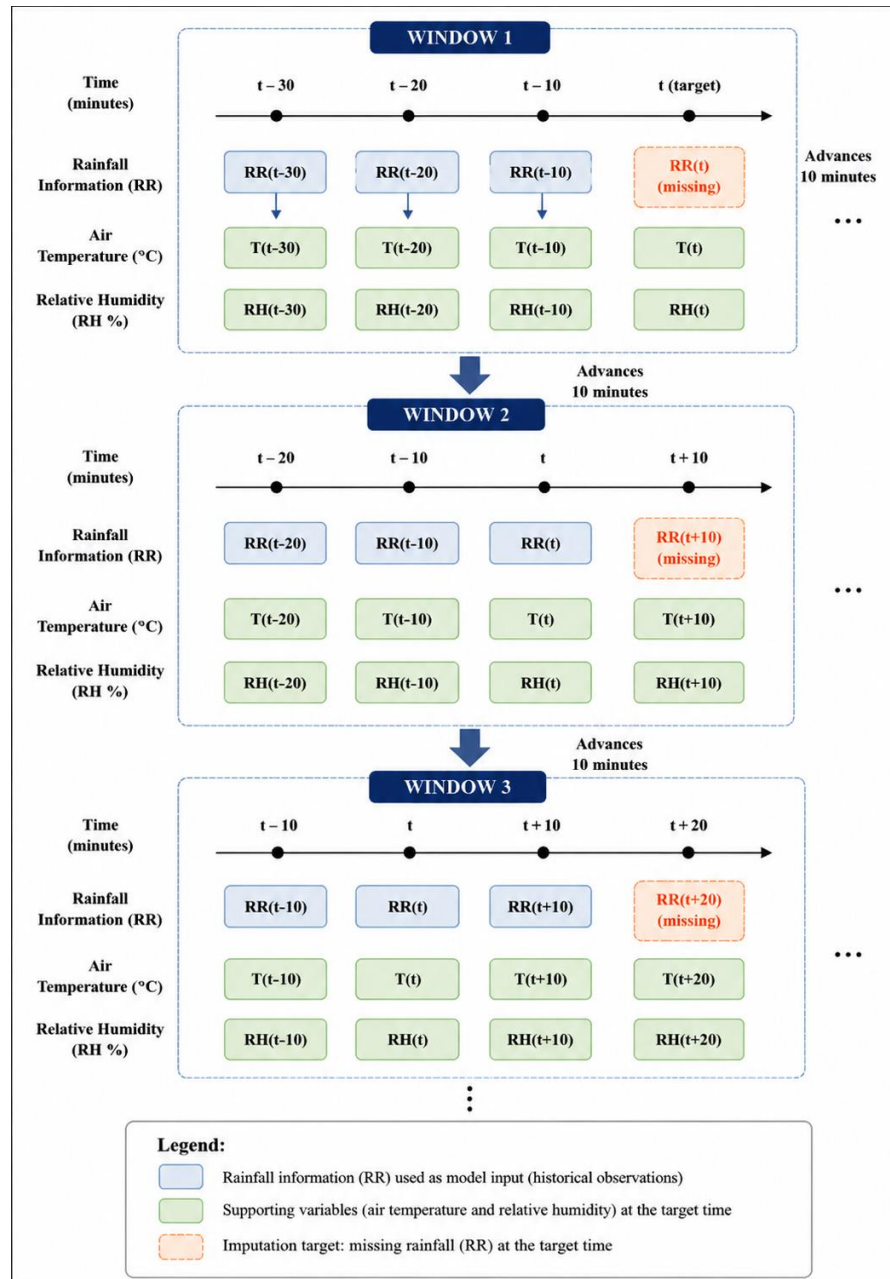


Figure 1. Illustration of the sliding-window sample construction in this research

Figure 1 shows that consecutive samples share most of their historical inputs. This overlap preserves local continuity and produces a large number of training examples without rearranging the time series. The 30-minute context was selected to capture immediate rainfall persistence and short convective changes; however, it also limits the amount of long-range information available to the recurrent model and is therefore considered when interpreting the comparative results

D. Chronological Partitioning and Scaling

The model-ready dataset was divided chronologically into 60% training, 20% validation, and 20% testing subsets. This time-based split prevents future observations from influencing model fitting and is more appropriate than random shuffling for sequential data. Table 2 presents the number of samples and the role of each subset.

Table 2. Chronological Dataset Partition

Subset	Share	Samples	Purpose
Training	60%	145,498	Parameter estimation
Validation	20%	48,499	Configuration selection and early stopping
Testing	20%	48,500	Final out-of-sample evaluation

Table 2 indicates that 145,498 samples were used for parameter estimation, 48,499 for configuration selection and early stopping, and 48,500 for final evaluation. Scaling parameters were estimated from the training subset only and then applied to validation and test data, thereby avoiding information leakage from later periods.

E. LSTM Model

The LSTM model used a compact sequence representation containing three rainfall lags and two contemporaneous meteorological predictors. A recurrent layer with 32 hidden units processed this information through forget, input, and output gates, after which a single regression output reconstructed $RR(t)$. This design tested whether gated memory could summarize immediate rainfall evolution from a 30-minute context while incorporating temperature and relative humidity.

The architectural components and training settings used for the LSTM experiments are summarized in Table 3. Three batch-size conditions were evaluated while the remaining model structure was held constant, allowing the influence of gradient-update frequency to be examined without changing the recurrent capacity.

Table 3. LSTM architecture and training configuration

Component	Configuration	Purpose
Input predictors	$RR(t-1)$, $RR(t-2)$, $RR(t-3)$, $T(t)$, and $RH(t)$	Represent the recent rainfall context and atmospheric state
Recurrent layer	LSTM with 32 hidden units	Learn gated temporal dependencies
Regression output	One continuous value: $RR(t)$	Reconstruct the missing rainfall observation
Optimizer	Adam; learning rate 0.001	Update network weights adaptively
Training scenarios	Batch sizes 32, 16, and 8; maximum 100 epochs	Evaluate sensitivity to update frequency
Early stopping	Patience 10, 15, and 15, respectively	Limit overfitting using validation performance

Table 3 separates the fixed architecture from the parameters varied during experimentation. The 32-unit recurrent layer and one-value regression head remained unchanged, whereas batch size and early-stopping patience defined the three tested conditions. This controlled arrangement ensured that changes in LSTM performance could be interpreted mainly as training-dynamics effects rather than as changes in model depth or hidden-state capacity.

F. Transformer Model

The Transformer received the same five predictors and target definition as the LSTM, but processed them through an encoder based on self-attention. Query, key, and value representations allowed the model to assign different relevance to the rainfall lags, temperature, and relative humidity without condensing all information through a recurrent hidden state. A regression head then mapped the encoded representation to one reconstructed rainfall value. Table 4 presents the Transformer architecture and the training configurations evaluated in the experiment. The input definition, encoder mechanism, and regression target were kept fixed, while batch size and early-stopping patience followed the same three-condition design used for the LSTM

Table 4. Transformer architecture and training configuration

Component	Configuration	Purpose
Input predictors	RR(t-1), RR(t-2), RR(t-3), T(t), and RH(t)	Provide temporal and meteorological information
Sequence-processing block	Transformer encoder with self-attention	Model direct relationships among the inputs
Attention representation	Query, key, and value transformations	Weight each input according to its relevance
Regression output	One continuous value: RR(t)	Reconstruct the missing rainfall observation
Training scenarios	Adam; batch sizes 32, 16, and 8; maximum 100 epochs	Apply comparable optimization conditions
Early stopping	Patience 10, 15, and 15, respectively	Select a stable model using validation performance
Input predictors	RR(t-1), RR(t-2), RR(t-3), T(t), and RH(t)	Provide temporal and meteorological information

As summarized in Table 4, the principal architectural distinction lies in the self-attention encoder, which evaluates relationships among the available inputs directly. The training budget was aligned with the LSTM experiment so that the comparison emphasized the difference between recurrent memory and attention-based representation rather than unequal optimization settings.

G. Evaluation Metrics

Model performance was evaluated using RMSE, MAE, and R^2 . RMSE gives greater weight to large deviations and is useful for detecting failures on higher rainfall values. MAE measures the typical absolute discrepancy in the original rainfall unit and is less dominated by a small number of large residuals. R^2 quantifies the proportion of observed variation represented by the reconstructed series. The three measures were calculated as follows:

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

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - (\bar{y}))^2} \quad (3)$$

where y_i is the observed rainfall, $\hat{y}_i$ is the reconstructed rainfall, $\bar{y}$ is the mean observed rainfall, and n is the number of evaluated samples. Lower RMSE and MAE indicate smaller errors, whereas an R^2 value closer to one indicates stronger preservation of observed variability. Because these metrics emphasize different aspects of performance, the preferred configuration was selected from their combined interpretation rather than from a single value

3. RESULT AND DISCUSSION

A. Dataset Characteristics

Before examining model accuracy, the distribution of the predictors and target was evaluated because it directly affects loss optimization and error interpretation. Table 4 summarizes the descriptive statistics of rainfall, temperature, and relative humidity in the model-ready dataset

Table 5. Descriptive statistics of the modeling variables

Variable	Mean	Minimum	Maximum
Rainfall, RR (mm/10 min)	0.052	0.00	48.20
Air temperature, T (°C)	27.94	21.77	38.86
Relative humidity, RH (%)	76.02	17.52	100.00

Table 5 reveals an extreme contrast between the mean rainfall of 0.052 mm/10 min and the maximum of 48.20 mm/10 min. The mean being close to zero indicates that dry observations dominate the series, whereas intense rain occupies a small part of the sample space. Under this distribution, a model can obtain a relatively low average loss by producing values near zero for many records, even if its performance on wet events is weaker. Consequently, aggregate RMSE and MAE must be interpreted together with R^2 and with the behavior shown in representative dry-period predictions.

Temperature ranged from 21.77 to 38.86 °C and relative humidity from 17.52% to 100%. These variables are smoother than rainfall and cannot uniquely determine whether precipitation occurs, but they provide atmospheric context that helps distinguish similar rainfall-lag patterns. High humidity, for example, may increase the plausibility of continuing rainfall, while the combination of dry recent lags and lower humidity may support a near-zero estimate. Their contribution is therefore conditional rather than deterministic.

B. Convergence Behavior

Training and validation loss were examined before comparing test metrics to determine whether the selected configurations had converged and whether a large generalization gap was present. Figure 2 displays the loss curves for the best LSTM and Transformer configurations, both of which used a batch size of 8.

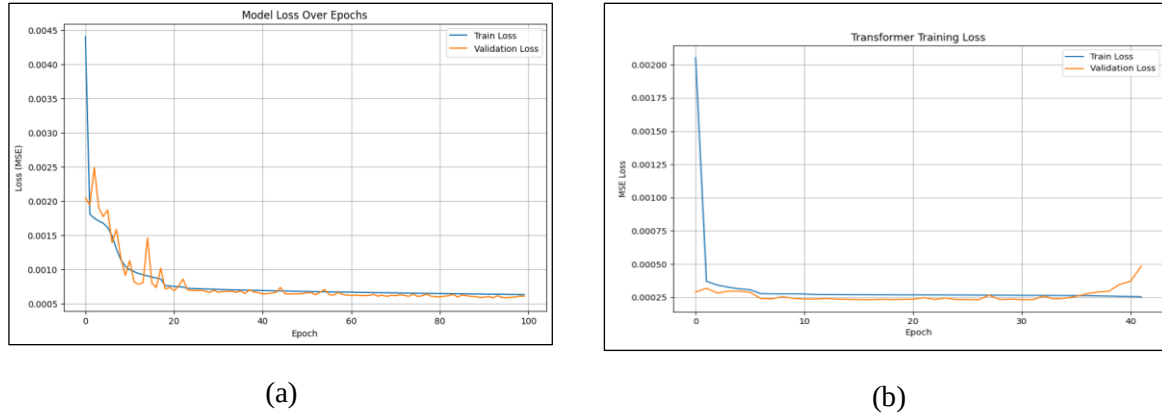


Figure 2. Learning Curve of (a)LSTM and (b) Transformer configuration

The LSTM curve declined rapidly during the initial epochs and then flattened, ending at a training loss of 0.000630 and a validation loss of 0.000606 after 100 epochs. The two curves remained close and did not show a sustained increase in validation loss, indicating stable optimization. The slightly lower validation loss can occur when training-time stochasticity or regularization makes the training objective marginally noisier than evaluation on the validation set; it does not by itself indicate data leakage.

The Transformer reached a lower training loss of 0.000258 and a validation loss of 0.0007519, with early stopping at epoch 42. Its training-validation gap was larger than that of LSTM, but the validation curve remained stable rather than diverging. Early stopping prevented further specialization to the training subset, and the later test results show that the retained parameters generalized well. Therefore, loss magnitude should be used mainly to assess convergence within each architecture, not to compare architectures directly, because their internal scaling and learning dynamics are different.

C. LSTM Performance

The effect of batch size on LSTM is summarized in Table 5. The table reports validation and test metrics for all three configurations, allowing optimization behavior and out-of-sample performance to be assessed together

Table 6. LSTM Validation and test performance

LSTM batch size	Validation RMSE	Validation MAE	Validation R ²	Test RMSE	Test MAE	Test R ²
32	1.3569	0.6412	0.6538	1.3420	0.6234	0.6542
16	1.2824	0.5951	0.6601	1.2676	0.5778	0.6575
8	1.1674	0.5301	0.6649	1.1520	0.5140	0.6626

Table 6 shows a monotonic improvement as batch size decreases. On the test set, RMSE falls from 1.3420 at batch size 32 to 1.1520 at batch size 8, a reduction of approximately 14.2%. MAE decreases from 0.6234 to 0.5140, or about 17.5%, while R² rises from 0.6542 to 0.6626. The larger proportional gain in MAE than in R² indicates that smaller batches mainly reduced the typical magnitude of errors but only modestly improved the representation of overall rainfall variability.

The validation and test values remain closely aligned across configurations. For batch size 8, the test RMSE is only 0.0154 lower than the validation RMSE, the test MAE is 0.0161 lower, and test R^2 is 0.0023 lower. These small differences suggest that the chosen configuration did not overfit the validation period and that the chronological test segment had a comparable level of difficulty.

Despite stable generalization, the best LSTM explains only about 66% of observed rainfall variance. The compact three-step sequence may have constrained the recurrent architecture because LSTM is most advantageous when it can accumulate information over a longer temporal horizon. In this experiment, the network repeatedly updates a memory state for only three rainfall lags, so part of its sequential capacity may remain underused. A longer window could improve event development representation, but it would also increase model complexity and reduce the number of complete training samples.

Operationally, LSTM remains attractive because its recurrent structure is familiar, comparatively straightforward to deploy, and capable of processing observations sequentially. However, an RMSE above 1 mm/10 min is substantial relative to the low mean rainfall, implying that the model's larger errors are concentrated in a smaller number of wet observations. This interpretation is consistent with the imbalance shown in Table 4 and motivates an event-stratified evaluation in future work.

D. LSTM Imputation Characteristics

Aggregate metrics do not show whether a model introduces false rainfall during dry conditions. To inspect this behavior, Figure 3 presents five dry observations from the test illustration and the corresponding estimates produced by the best LSTM configuration.

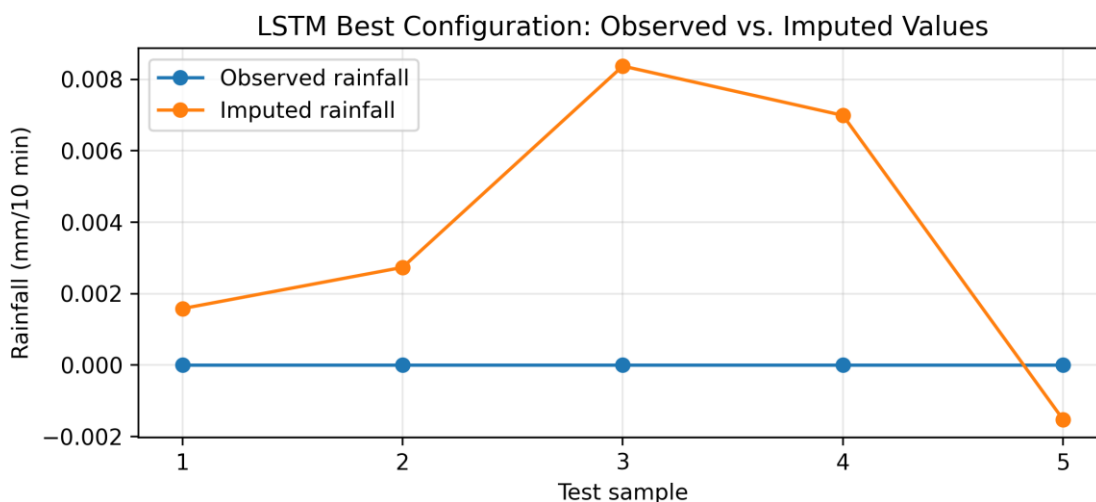


Figure 3. Comparison between actual and predict data on LSTM model

Figure 3 shows estimates between approximately -0.0015 and 0.0084 mm for observations whose actual rainfall is 0 mm. These values are very close to the dry target and demonstrate that the best LSTM can preserve no-rain conditions for the selected samples. Nevertheless, the illustration covers only five cases and should not be treated as a substitute for the 48,500-sample test metrics in Table 6.

The small negative estimate results from an unconstrained regression output and has no physical interpretation. In an operational pipeline, negative predictions should be clipped to zero or prevented by a nonnegative activation such as Softplus. A trace-rain threshold may also be applied, but it should be calibrated against station measurement resolution to avoid erasing genuine light rainfall.

LSTM processes the three rainfall lags in order, and the final hidden representation must summarize their joint temporal information before the regression layer estimates rainfall. This mechanism is useful when temporal order is informative, but with a short window it may compress the same limited evidence that is available explicitly to the attention model. The weaker aggregate R^2 therefore appears more related to representation efficiency than to optimization failure, because the loss curves and validation-test gaps indicate successful training.

Target imbalance further shapes LSTM behavior. Standard squared-error training receives many updates from dry intervals and relatively few from intense rainfall. Although large wet-event residuals receive a stronger quadratic penalty, the network may still learn a conservative output distribution. Future experiments should report separate dry-event, light-rain, moderate-rain, and heavy-rain metrics or use a weighted objective to determine whether improved overall accuracy also preserves extremes.

E. Transformer Performance

Transformer performance across batch sizes is reported in Table 6. As in the LSTM experiment, the validation metrics were used during model development and the chronological test metrics were retained for final assessment.

Table 7. Transformer Performance Model

Transformer batch size	Validation RMSE	Validation MAE	Validation R^2	Test RMSE	Test MAE	Test R^2
32	0.8596	0.3818	0.8118	0.8455	0.3720	0.8247
16	0.8448	0.3762	0.8144	0.8303	0.3666	0.8238
8	0.8464	0.3762	0.8167	0.8287	0.3635	0.8244

Table 7 shows that reducing batch size from 32 to 8 lowers test RMSE from 0.8455 to 0.8287 and MAE from 0.3720 to 0.3635, corresponding to improvements of about 2.0% and 2.3%. Test R^2 changes only marginally, from 0.8247 at batch size 32 to 0.8244 at batch size 8. This small trade-off illustrates why configuration selection should consider multiple metrics: batch size 8 minimizes absolute error, whereas batch size 32 produces a numerically higher R^2 by 0.0003, a difference too small to indicate a meaningful practical advantage.

The batch-size-8 Transformer stops after 42 epochs, compared with 19 and 26 epochs for batch sizes 32 and 16. Smaller batches introduce noisier but more frequent gradient updates and therefore require more iterations before the validation criterion stabilizes. The final test RMSE is 0.0177 lower than validation RMSE, test MAE is 0.0127 lower than validation MAE, and test R^2 is 0.0077 higher, demonstrating that early stopping selected a configuration that generalized at least as well to the later test period.

The stronger performance can be related to the self-attention mechanism, which evaluates each rainfall lag and meteorological predictor directly for the target estimate. With only three temporal lags, avoiding recurrent compression may be advantageous: the model can assign high relevance to the most informative recent observation while still considering humidity and temperature interactions. This interpretation is consistent with studies reporting competitive Transformer performance on multivariate and incomplete time series [13]–[17], but it remains specific to the current station and input window.

F. Transformer Imputation Characteristics

To complement the aggregate metrics, Figure 5 shows the same type of dry-period illustration for the batch-size-8 Transformer. The figure is used to examine output bias near the physical lower bound of rainfall.

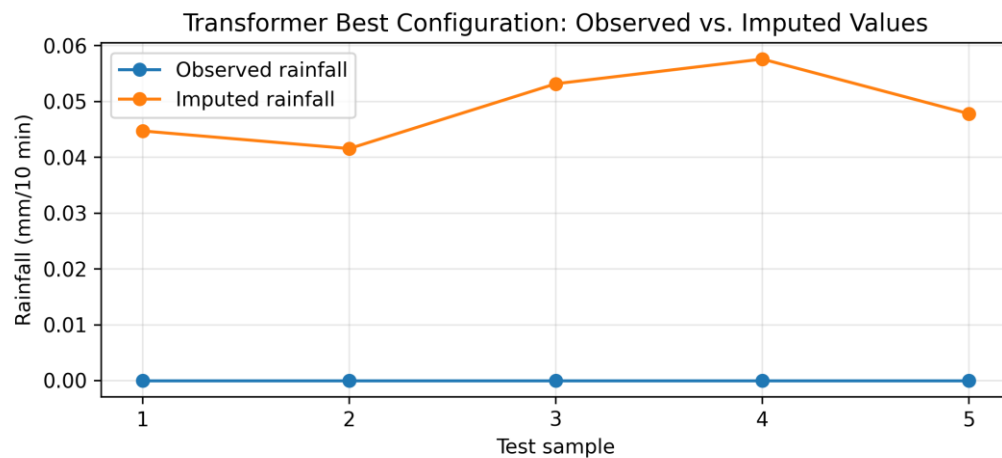


Figure 4. Comparison between actual and predict data on Transformer model

Figure 4 shows positive estimates of approximately 0.042-0.058 mm for five observations with actual rainfall of 0 mm. On this small subset, the Transformer has a greater dry bias than the best LSTM, even though it performs substantially better over the complete test set. The contrast demonstrates that a handful of visual samples can emphasize one local behavior and should always be interpreted together with aggregate results.

The positive dry bias can be reduced through station-specific post-processing, such as mapping predictions below the instrument's trace threshold to zero. However, thresholding must be evaluated separately for dry and light-rain cases because an aggressive rule could improve false-rain frequency while suppressing real drizzle. A nonnegative output layer would address physical validity but would not by itself remove small positive bias.

G. Comparison of Best Configurations Model

The best configurations are compared directly in Figure 6 to quantify the architecture-level difference under the same test set. The figure presents RMSE and MAE as error measures and R^2 as a measure of explained variation.

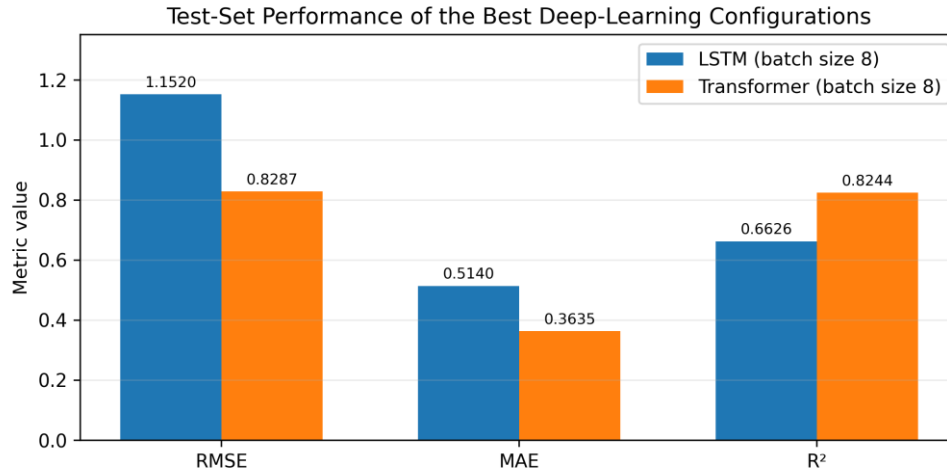


Figure 5. Performance Comparison between LSTM and Transformer model

Figure 6 confirms that the Transformer is superior on all three selected test metrics. Relative to the best LSTM, it reduces RMSE by 0.3233 mm, or approximately 28.1%, and MAE by 0.1505 mm, or approximately 29.3%. R^2 increases from 0.6626 to 0.8244, an absolute gain of 0.1618. The simultaneous reduction in both error measures and increase in explained variance indicates that the advantage is not caused by one isolated metric.

The error reduction is meaningful in relation to the data distribution. Because the average rainfall is only 0.052 mm/10 min, both models are evaluated in a dataset where most targets are close to zero. A model can therefore appear accurate on dry intervals while still missing wet-event magnitude. The Transformer's higher R^2 suggests that its benefit extends beyond dry-value approximation and includes a stronger ability to reproduce variability across the complete test series.

The comparative outcome also clarifies the role of the short input window. LSTM is designed to maintain information over long sequences, but the experiment provides only three temporal rainfall steps. Transformer attention can use the same compact representation without sequentially compressing it, which may explain why it extracts more useful information from the available predictors. This finding strongly aligns with the state-of-the-art research by Zerveas et al. [14] and Lim et al. [13], who established that attention-based architectures consistently outperform recurrent models in capturing complex multivariate dependencies without losing context. Furthermore, while previous studies (e.g., Ahn et al. [17]) highlighted LSTM's strength in long-term memory scenarios, our empirical results reveal a crucial insight: for highly imbalanced, short-duration convective rainfall data, the Transformer's ability to directly weigh recent lags without sequential bottlenecking provides a superior reconstruction capability. This does not imply that Transformers are universally superior; a longer sequence, different masking pattern, or larger spatial dataset could change the ranking. The dry-sample figures reveal a complementary trade-off. LSTM produces values closer to zero in the selected dry cases, whereas Transformer produces lower aggregate errors and a much higher R^2 . Model selection should therefore prioritize the complete evaluation while retaining a post-processing rule for physical consistency. For operational use, the selected model should also be monitored with event-based diagnostics so that performance during heavy rain is not obscured by the dominant dry class.

4. CONCLUSION

This study compared LSTM and Transformer architectures for imputing missing 10-minute rainfall observations using three recent rainfall lags, air temperature, and relative humidity from the South Tangerang AWS. Both models converged and generalized consistently, but the Transformer produced

the stronger overall reconstruction. The best LSTM configuration used a batch size of 8 and achieved an RMSE of 1.1520, an MAE of 0.5140, and an R^2 of 0.6626, whereas the best Transformer configuration, also using a batch size of 8, achieved an RMSE of 0.8287, an MAE of 0.3635, and an R^2 of 0.8244. These results correspond to reductions of approximately 28.1% in RMSE and 29.3% in MAE, together with an absolute R^2 improvement of 0.1618. The Transformer is therefore recommended for the studied dataset, provided that operational deployment includes nonnegative post-processing, trace-rain calibration, imputation flags, and periodic model monitoring. Further validation should examine longer missing blocks, rainfall-intensity-specific performance, longer temporal windows, and transfer to additional tropical AWS locations.

ACKNOWLEDGEMENT

The authors acknowledge the Indonesian Agency for Meteorology, Climatology, and Geophysics and the Banten Climatological Station for providing the Automatic Weather Station observations used in this research. The authors also acknowledge the Master of Informatics Engineering Program, Universitas Pamulang, for academic support.

REFERENCES

- [1] World Meteorological Organization. (2024). Guide to instruments and methods of observation (WMO-No. 8). Geneva: WMO.
- [2] World Meteorological Organization. (2023). Guide to climatological practices. Geneva: WMO.
- [3] World Meteorological Organization. (2021). Manual on the WMO Integrated Global Observing System: Annex VIII to the WMO Technical Regulations. Geneva: WMO.
- [4] Calvin, K., Dasgupta, D., Krinner, G., Mukherji, A., Thorne, P. W., Trisos, C., et al. (2023). Climate Change 2023: Synthesis Report. Geneva: Intergovernmental Panel on Climate Change. <https://doi.org/10.59327/IPCC/AR6-9789291691647.001>
- [5] Little, R. J. A., & Rubin, D. B. (2019). Statistical analysis with missing data (3rd ed.). Hoboken, NJ: Wiley.
- [6] Alejo-Sanchez, L. E., Márquez-Grajales, A., Salas-Martínez, F., Franco-Arcega, A., López-Morales, V., Acevedo-Sandoval, O. A., et al. (2025). Missing data imputation of climate time series: A review. *MethodsX*, 15, 103455. <https://doi.org/10.1016/j.mex.2025.103455>
- [7] Addi, M., Gyasi-Agyei, Y., Obuobie, E., & Amekudzi, L. K. (2022). Evaluation of imputation techniques for infilling missing daily rainfall records on river basins in Ghana. *Hydrological Sciences Journal*, 67(4), 613-627. <https://doi.org/10.1080/02626667.2022.2030868>
- [8] Wangwongchai, A., Waqas, M., Dechpichai, P., Hlaing, P. T., Ahmad, S., & Humphries, U. W. (2023). Imputation of missing daily rainfall data: A comparison between artificial intelligence and statistical techniques. *MethodsX*, 11, 102459. <https://doi.org/10.1016/j.mex.2023.102459>
- [9] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- [10] Zhang, Y., Ragetti, S., Molnar, P., Fink, O., & Peleg, N. (2022). Generalization of an encoder-decoder LSTM model for flood prediction in ungauged catchments. *Journal of Hydrology*, 614, 128577.
- [11] Khairunnisa, A., Sartono, B., & Aidi, M. N. (2025). Missing data imputation for LSTM-based flood early warning system in Jakarta. *International Research Journal of Innovations in Engineering and Technology*, 9(6), 142-148. <https://doi.org/10.47001/irjiet/2025.906018>
- [12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

- [13] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748-1764. <https://doi.org/10.1016/j.ijforecast.2021.03.012>
- [14] Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A Transformer-based framework for multivariate time series representation learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2114-2124. <https://doi.org/10.1145/3447548.3467401>
- [15] Du, W., Wang, J., Qian, L., Yang, Y., Ibrahim, Z., Liu, F., et al. (2024). TSI-Bench: Benchmarking time series imputation. *arXiv:2406.12747*.
- [16] Atmaca, K., & Yenilmez, İ. (2024). RNNs and Transformer model in case of incomplete time series. *International Applied Statistics Congress 2024, Istanbul, Turkey*.
- [17] Ahn, H., Sun, K., & Kim, K. P. (2022). Comparison of missing data imputation methods in time series forecasting. *Computers, Materials & Continua*, 70(1), 767-779. <https://doi.org/10.32604/cmc.2022.019369>
- [18] Alruhaymi, A. Z., & Kim, C. J. (2021). Study on the missing data mechanisms and imputation methods. *Open Journal of Statistics*, 11(4), 477-492. <https://doi.org/10.4236/ojs.2021.114030>
- [19] Alwateer, M., Atlam, E.-S., El-Raouf, M. M. A., Ghoneim, O. A., & Gad, I. (2024). Missing data imputation: A comprehensive review. *Journal of Computer and Communications*, 12(11), 53-75. <https://doi.org/10.4236/jcc.2024.1211004>
- [20] Li, C., Ren, X., & Zhao, G. (2023). Machine-learning-based imputation method for filling missing values in ground meteorological observation data. *Algorithms*, 16(9), 422. <https://doi.org/10.3390/a16090422>