



Reconstruction of Missing Meteorological Observation Data Using a Hybrid LSTM-KNN Model in a Web-Based Monitoring System

Muhammad Rafi Athallah Disastra^{1*}, Edward Trihadi¹, Nardi¹, Marzuki Sinambela¹

¹ Department of Instrumentation-MKG, Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, Tangerang, Banten 15119, Indonesia

DOI: <https://doi.org/10.29303/goescienceed.v7i4.3215>

Article Info:

Received : August, 27 2026
Revised : September, 15 2026
Accepted : September, 27 2026
Published : October, 13 2026

Correspondence:

Muhammad Rafi Athallah Disastra

Phone:

Abstract: Automatic Weather Stations frequently lose data because of sensor faults, power failures, or network disruption, which lowers the validity of observational records. This study develops a hybrid Long Short-Term Memory (LSTM) and K-Nearest Neighbors (KNN) model to reconstruct missing data and implements it in a web-based system. KNN acts as the initial estimator based on seasonal analogues, while LSTM refines the temporal pattern. Both estimates are merged through a dynamic weighted blending mechanism that decays with gap duration to prevent autoregressive drift, complemented by buffer isolation so the LSTM never reads its own predictions as context. The system was built with Streamlit and SQLite, covering layered quality control, reconstruction, and a monitoring dashboard. Evaluation used nine artificial gap scenarios on AWS data from Tanjung Priok Station for 2020 to 2026, covering air temperature, relative humidity, and air pressure. The hybrid advantage proved conditional on gap structure and duration. On block gaps it lowered MAE by up to 2.66% over KNN-only, peaking at ten minutes and shrinking to 0.07% at six hours, whereas on sporadic MCAR gaps it gave no advantage. The findings show that adaptive weight decay is essential in operational systems.

Keywords: AWS; Missing Data; Hybrid LSTM-KNN; Data Reconstruction; Dynamic Weighted Blending.

Citation: Disastra, M. R. A., Trihadi, E., Nardi, & Sinambela, M. (2026). Reconstruction of Missing Meteorological Observation Data Using a Hybrid LSTM-KNN Model in a Web-Based Monitoring System. *Jurnal Pendidikan, Sains, Geologi, Dan Geofisika (GeoScienceEd Journal)*, 7(4), 6184–6191. <https://doi.org/10.29303/goescienceed.v7i4.3215>

Introduction

Automatic Weather Stations (AWS) are standard meteorological instruments that measure weather parameters automatically and continuously (World Meteorological Organization, 2023). An AWS comprises integrated sensors, a data processing unit, and a communication system that transmits data in real time to a central server (Sugiarto, Rahmat, & Purnamawati, 2019). Although designed to operate autonomously, an AWS is not immune to technical problems that cause data loss, such as sensor damage, power supply failure, and network disruption during transmission (Wahyudi & Widyarto, 2025), as well as extreme environmental conditions (Boomgard-Zagrodnik & Brown, 2022). Unhandled missing data directly reduces the validity of

measurement results and may lead to biased analysis (Nida, Kashif, Khan, & Ghamkhar, 2023).

Various imputation approaches have been developed for high-resolution climate time series (Afrifa-Yamoah, Mueller, Taylor, & Fisher, 2020), and several studies have compared statistical and machine learning methods (Ahn, Sun, & Kim, 2022). The K-Nearest Neighbors (KNN) algorithm fills missing values from the k nearest neighbours under a chosen distance measure and has proven effective on meteorological agency data (Oktaviani & Putrada, 2022), with continuing development for time series (Almeida, Bras, Sargento, & Pinto, 2024). For meteorological records, similarity can be defined temporally and seasonally by searching for historical observations at a similar day-of-

Email: rafi.athallah1@gmail.com

year and minute-of-day position. Long Short-Term Memory (LSTM), an extension of the recurrent neural network that retains long-term information through a gating mechanism, excels at modelling the sequential dependence of weather data (Lattifia, Buana, & Rusjayanthi, 2022; Nugraha, Ariawan, & Arifin, 2023).

Combining the two in a hybrid scheme offers advantages over a single method, with KNN providing an initial estimate from pattern similarity and LSTM aligning the result with the temporal trend (Luo, Li, Yang, & Zhang, 2019; Liu et al., 2020). LSTM-based hybrids also show promise for multivariate environmental imputation (Laeq, Li, & Adeel, 2026), and applying imputation before further analysis improves LSTM output quality (Nugroho & Haris, 2024). Nevertheless, most of these studies evaluate models under uniform gap conditions and report hybrid superiority only as an aggregate figure, without testing whether the advantage holds, weakens, or reverses as gap structure and duration change. This leaves an open question of practical consequence: whether a hybrid KNN-LSTM scheme can be trusted under a single fixed weighting across all missing-data conditions, or whether its benefit is conditional and must be actively managed. To the authors' knowledge, prior work in this line has not explicitly mapped that boundary, nor proposed a mechanism that adapts the blending weight to gap duration while safeguarding the LSTM from the autoregressive drift such long reconstructions induce. This gap motivates the present study, which contributes a gap-duration-aware blending mechanism together with an explicit evaluation of where the hybrid advantage begins, peaks, and dissolves.

A further consideration is that prediction executed autoregressively feeds the model output at one step back as the input for the next. Small errors accumulate, so predictions deviate progressively as the span lengthens (Ouyang, Ravier, & Jabloun, 2022; Cerqueira & Torgo, 2023). This phenomenon has two distinct failure points, and each is addressed by a different mechanism in the proposed system. First, trusting the LSTM equally regardless of how long a parameter has been missing lets its compounding error dominate the estimate on long gaps; this motivates a damping mechanism, analogous to the damped-trend principle in exponential smoothing (Hyndman & Athanasopoulos, 2021), that progressively lowers the LSTM weight as the gap lengthens. Second, if the model output is written back into the buffer it consults, the LSTM ends up conditioning on its own earlier predictions rather than on observed data, a training-inference discrepancy known as exposure bias (Teutsch & Mäder, 2022), which is the direct mechanical cause of autoregressive drift; this is addressed separately through buffer isolation, which keeps the buffer

populated with the stable KNN estimate instead of the blended output. On the operational side, manual validation is inefficient for continuous AWS operation because it is time consuming and prone to human error (Purwandari, Aufauzan, & Sigalingging, 2026), whereas modern frameworks such as Streamlit allow complex models to be packaged into interactive dashboards (Akkem, Biswas, & Varanasi, 2024).

This study therefore pursues three objectives. The first is to design a dynamic weighted blending mechanism—a composite estimate that combines the KNN and LSTM outputs through a weight that shifts between them according to how long a parameter has been continuously missing, following the principle of dynamic forecast combination in which weights adapt over time (Cerqueira & Torgo, 2023). The weight follows a damping schedule, analogous to the damped-trend principle in exponential smoothing (Hyndman & Athanasopoulos, 2021), meaning it decays progressively as the gap lengthens so that confidence in the LSTM's temporal extrapolation is reduced the longer it operates without fresh observations. This is complemented by buffer isolation, a safeguard that always feeds the LSTM's rolling input buffer with the stable KNN estimate rather than the blended output, preventing the model from treating its own prior predictions as observed data and thereby stopping chained prediction error from accumulating (Ouyang, Ravier, & Jabloun, 2022; Teutsch & Mäder, 2022). The second is to evaluate the model across nine gap scenarios of varying structure and duration so that the boundary of the hybrid advantage can be mapped explicitly. The third is to implement the model in an operational web system integrated with a database. The scope is limited to three continuous parameters, namely air temperature, relative humidity, and air pressure. Rainfall is excluded because its data are intermittent and dominated by zero values, which requires a two-stage approach beyond the present scope.

Method

Data and Preprocessing

The data are AWS observations from Tanjung Priok Station for 1 January 2020 to 30 April 2026, accessed through a REST API (Ehsan, Abuhaliqa, Catal, & Mishra, 2022). The initial dataset contained 2,272,635 rows at non-uniform intervals. Time reindexing rearranged the index to be continuous at one-minute resolution, producing 3,329,233 rows. Quality control applied a range check, a step check, and a persistence check, after which all values were normalized using min-max scaling. Because the persistence check requires a long history window, it is applied when constructing the historical reference database, while the operational path applies the range and step checks that can be

evaluated instantaneously. Table 1 lists the thresholds used. The data were divided into training and operational subsets, and evaluation was performed only on the operational subset, which the model never saw during training.

Table 1. Quality control thresholds per parameter

Parameter	Unit	Valid range	Max. step
Air temperature	°C	17.0 - 42.0	1.5
Relative humidity	%	15.0 - 100.0	4.0
Air pressure	hPa	985.0 - 1020.0	1.0

System Workflow

The overall workflow is shown in Figure 1. Data are acquired from the API and immediately pass through quality control. When no missing or invalid value is detected, the flow proceeds directly to the rolling buffer update without invoking the reconstruction models, preserving computational efficiency. When a missing or invalid value is found, the system first computes the KNN estimate, then the LSTM prediction, and combines the two through dynamic weighted blending. The result is denormalized, stored in an SQLite database (Gaffney et al., 2022), and displayed on a dashboard built with Streamlit (Akkem et al., 2024).

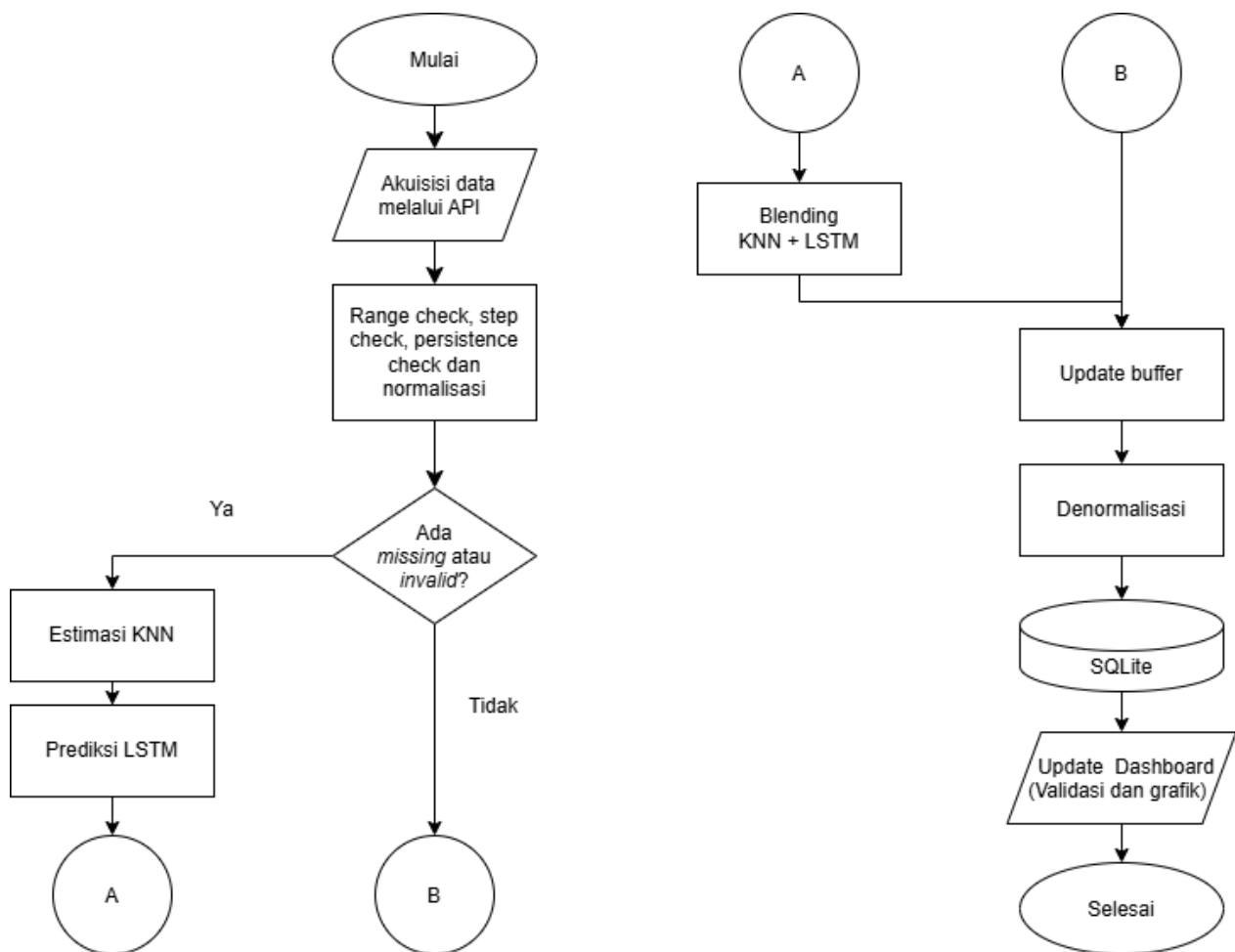


Figure 1. Flowchart of the AWS data reconstruction system

KNN and LSTM Components

The KNN component estimates missing values by matching seasonal analogue patterns using circular distance on day-of-year and minute-of-day. The search window is widened progressively when no neighbour is found. The estimate is computed as a weighted average, with weights inversely proportional to the rank of proximity, as shown in (1). where $\hat{x}$ is the estimated value, x_i the value of the i -th neighbour, and w_i the

normalized weight. The climatological estimate is then corrected with an anchor offset, namely the difference between the last valid reading before the gap and the KNN estimate at that same instant, which decays gradually according to the physical characteristics of each parameter.

$$\hat{x} = \sum w_i x_i, \quad w_i = \left(\frac{1}{r_i}\right) / \sum \left(\frac{1}{r_j}\right) \quad (3)$$

The LSTM component produces an independent prediction from a rolling buffer of historical data. Hyperparameter optimization used grid search for KNN over 20 combinations and random search for LSTM over

10 sampled configurations, the latter being more efficient for large and partly continuous spaces (Yang & Shami, 2020). Table 2 summarizes the search space and the selected values.

Table 2. Hyperparameter search space and selected values

Model	Hyperparameter	Search space	Selected
KNN	Number of neighbours k	3, 5, 7, 10, 15	5
KNN	Search window (days)	15, 30, 45, 60	60
LSTM	Input window (minutes)	30, 60, 120	120
LSTM	Units per layer	32, 64, 128	128
LSTM	Number of layers	1, 2, 3	2
LSTM	Dropout rate	0.1, 0.2, 0.3	0.1
LSTM	Learning rate	0.0001, 0.0005, 0.001	0.0005
LSTM	Batch size	64, 128, 256	128

Dynamic Weighted Blending and Buffer Isolation

The two estimates are combined through dynamic weighted blending as expressed in (2).

$$\hat{x} = \alpha(s) \hat{x}_{\text{LSTM}} + (1 - \alpha(s)) \hat{x}_{\text{KNN}} \quad (2)$$

where s denotes how long a parameter has been reconstructed consecutively and $\alpha(s)$ is the LSTM weight. The weight follows a three-stage curve: rising from 0.15 to 0.55 during the first ten minutes, decaying towards 0.15 until minute 90, then decaying further towards a floor of 0.02, as plotted in Figure 2. The blended result is capped against the last valid reading and against the previous minute's reconstruction to prevent implausible jumps.

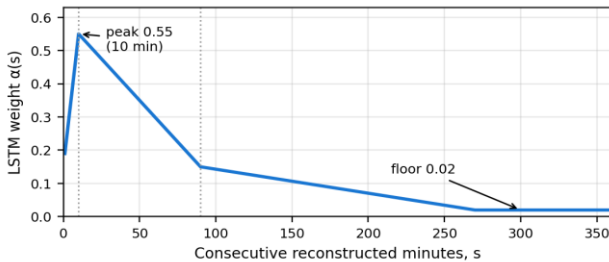


Figure 2. LSTM weight $\alpha(s)$ as a function of consecutive reconstructed minutes

The initial ramp allows the LSTM to contribute meaningfully once a gap is confirmed, the peak at ten minutes coincides with the range in which its temporal context remains valid, and the two decay stages progressively hand control back to KNN as the context degrades. The second mechanism is buffer isolation. If the blended result were fed directly into the buffer, the LSTM would read its own prediction as context in the following minute, causing autoregressive drift. To prevent this, the buffer is filled with the pure KNN

value, which is stable because it is grounded in historical references, while the value displayed and stored remains the blended result.

Evaluation Design

Evaluation used nine artificial gap scenarios, consisting of four Missing Completely At Random (MCAR) scenarios with loss rates of 5%, 10%, 20%, and 30%, and five block-missing scenarios of 5 minutes, 10 minutes, 30 minutes, 1 hour, and 6 hours. Each block scenario was repeated at random non-overlapping positions, and all gap points from every repetition were pooled into a single sample before the metrics were computed, so the result does not depend on one incidental location. Artificial gap positions were excluded from the KNN neighbour pool to prevent data leakage.

Three complementary metrics were used. Mean Absolute Error (MAE) in (3) measures the average magnitude of deviation in the original physical unit, Root Mean Square Error (RMSE) in (4) penalizes large deviations more heavily, and the coefficient of determination R^2 in (5) expresses the proportion of variance explained.

$$\text{MAE} = \left(\frac{1}{n} \right) \sum |y_i - \hat{y}_i| \quad (3)$$

$$\text{RMSE} = \sqrt{\left(\frac{1}{n} \right) \sum (y_i - \hat{y}_i)^2} \quad (4)$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad (5)$$

where y_i is the ground-truth value at the artificially removed point, $\hat{y}_i$ the reconstructed value, $\bar{y}$ the mean of the ground truth, and n the number of evaluated points. All values are denormalized to their physical units before the metrics are computed.

Result and Discussion

Hyperparameter Tuning Results

The eight best KNN configurations are listed in Table 3. The lowest MAE was obtained with $k = 5$ and a 60-day search window, which was therefore adopted. The ranking indicates that a wider seasonal window benefits the analogue search, since a narrow window frequently fails to supply enough contextual neighbours at the same minute-of-day.

Table 4 reports the ten LSTM configurations sampled by random search. The best validation loss was reached at iteration nine with a 120-minute window, two layers of 128 units, a dropout of 0.1, a learning rate of 0.0005, and a batch size of 128. Configurations using a 30-minute window consistently underperformed,

confirming that a longer context window is required to capture the diurnal pattern.

Tabel 3. Eight best KNN grid search configurations

Rank	k	Window (days)	MAE	RMSE	Selected
1	5	60	0.002533	0.009711	Yes
2	3	30	0.003122	0.009458	-
3	10	45	0.003279	0.011443	-
4	7	30	0.004343	0.012772	-
5	5	45	0.004472	0.013743	-
6	15	15	0.005008	0.015055	-
7	3	15	0.005406	0.017857	-
8	3	60	0.005503	0.015423	-

Tabel 4. LSTM random search results

Iter.	Window (min)	Units	Layers	Dropout	Learning rate	Batch	val_loss (MSE)
1	30	128	2	0.2	0.0005	256	0.000048
2	30	128	1	0.1	0.0005	256	0.000041
3	120	128	2	0.3	0.0005	64	0.000036
4	120	64	2	0.2	0.001	256	0.000043
5	120	64	3	0.3	0.0005	128	0.000038
6	60	32	1	0.2	0.0001	64	0.000048
7	120	128	1	0.3	0.001	256	0.000035
8	120	64	1	0.3	0.0005	256	0.000056
9	120	128	2	0.1	0.0005	128	0.000030
10	60	32	2	0.1	0.0001	256	0.000107

Reconstruction Performance

A summary of performance across all scenarios is presented in Table 5, where values are averaged over the

three parameters. The last column shows the percentage reduction in MAE of the hybrid relative to KNN-only; a positive value means the hybrid is superior.

Tabel 5. Performance comparison of KNN-only and hybrid KNN-LSTM

Scenario	MAE KNN	MAE Hyb	RMSE KNN	RMSE Hyb	R ² KNN	R ² Hyb	Δ MAE (%)
MCAR 5%	0.1387	0.1387	0.2333	0.2333	0.9979	0.9979	+0.00
MCAR 10%	0.1410	0.1411	0.2427	0.2428	0.9978	0.9978	-0.07
MCAR 20%	0.1470	0.1479	0.2522	0.2562	0.9976	0.9974	-0.59
MCAR 30%	0.1532	0.1567	0.2606	0.2719	0.9974	0.9970	-2.26
Block 5 min	0.3650	0.3607	0.4748	0.4699	0.9862	0.9865	+1.17
Block 10 min	0.4547	0.4426	0.6326	0.6156	0.9753	0.9765	+2.66
Block 30 min	0.9394	0.9255	1.4052	1.3884	0.8773	0.8801	+1.48
Block 1 h	1.5679	1.5533	2.1501	2.1334	0.7194	0.7237	+0.93
Block 6 h	3.0823	3.0802	4.0900	4.0889	0.2801	0.2803	+0.07

In the MCAR scenarios, both methods produce practically identical performance with R^2 above 0.997. The hybrid even lags slightly as the loss rate rises, reaching -2.26% at MCAR 30%. This is explained by the gap structure: MCAR gaps are almost entirely one minute long and isolated, so they do not reach the minimum duration threshold that activates the LSTM, while at high loss rates the pool of historical neighbours also thins out.

A different pattern appears in the block-missing scenarios. The hybrid is consistently superior across all block durations, peaking at a 2.66% MAE reduction for the 10-minute gap, then declining to 1.48% at 30 minutes, 0.93% at 1 hour, and only 0.07% at 6 hours, as visualized in Figure 3.

Figure 3(a) shows that reconstruction difficulty increases sharply with gap duration, with MAE rising from 0.3607 at a 5-minute gap to 3.0802 at 6 hours, while

R^2 falls from 0.9865 to 0.2803. This decline applies to both methods, indicating a fundamental limitation of reconstructing long gaps rather than a weakness of either algorithm.

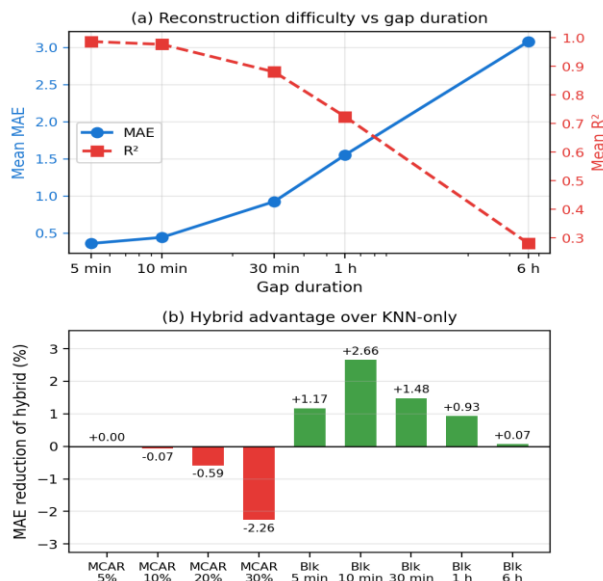


Figure 3. Reconstruction difficulty against gap duration

(a) and MAE difference of the hybrid relative to KNN-only (b)

Figure 3(b) clarifies that the hybrid advantage forms a curved pattern, largest at short-to-medium durations and then shrinking towards zero for long gaps. This behaviour matches the design intent: the weight decay reduces the LSTM contribution as the gap lengthens, so the estimate gradually leans on the more stable KNN. Convergence towards KNN-only performance at the 6-hour gap shows that the decay mechanism successfully prevents autoregressive drift, because without decay an LSTM prediction running autoregressively for hundreds of minutes could deviate substantially and worsen the result.

Per-parameter detail is presented in Table 6. Air pressure benefits most from the hybrid, with an MAE reduction of up to 6.77% at the 10-minute gap, followed by humidity and temperature. Pressure has persistent, smoothly varying characteristics, so its temporal pattern is easier for the LSTM to capture. Humidity, by contrast, shows the lowest R^2 at long gaps, namely 0.0048 at the 6-hour gap, because its fluctuation is the highest among the three parameters.

Table 6. Per-parameter performance detail

Scenario	Parameter	MAE KNN	MAE Hyb	R^2 KNN	R^2 Hyb	Δ MAE (%)
MCAR 5%	Temperature ($^{\circ}$ C)	0.0395	0.0395	0.9977	0.9977	+0.00
MCAR 5%	Humidity (%)	0.3632	0.3632	0.9962	0.9962	+0.00
MCAR 5%	Pressure (hPa)	0.0134	0.0134	0.9998	0.9998	+0.00
MCAR 10%	Temperature ($^{\circ}$ C)	0.0407	0.0408	0.9978	0.9978	-0.25
MCAR 10%	Humidity (%)	0.3685	0.3687	0.9958	0.9958	-0.05
MCAR 10%	Pressure (hPa)	0.0139	0.0139	0.9997	0.9997	+0.00
MCAR 20%	Temperature ($^{\circ}$ C)	0.0430	0.0434	0.9975	0.9972	-0.93
MCAR 20%	Humidity (%)	0.3831	0.3852	0.9955	0.9954	-0.55
MCAR 20%	Pressure (hPa)	0.0149	0.0150	0.9997	0.9997	-0.67
MCAR 30%	Temperature ($^{\circ}$ C)	0.0457	0.0471	0.9971	0.9962	-3.06
MCAR 30%	Humidity (%)	0.3981	0.4065	0.9953	0.9950	-2.11
MCAR 30%	Pressure (hPa)	0.0159	0.0165	0.9997	0.9997	-3.77
Block 5 min	Temperature ($^{\circ}$ C)	0.1182	0.1172	0.9918	0.9920	+0.85
Block 5 min	Humidity (%)	0.9259	0.9167	0.9676	0.9682	+0.99
Block 5 min	Pressure (hPa)	0.0508	0.0482	0.9991	0.9992	+5.12
Block 10 min	Temperature ($^{\circ}$ C)	0.1598	0.1560	0.9827	0.9834	+2.38
Block 10 min	Humidity (%)	1.1232	1.0962	0.9466	0.9490	+2.40
Block 10 min	Pressure (hPa)	0.0812	0.0757	0.9967	0.9972	+6.77
Block 30 min	Temperature ($^{\circ}$ C)	0.3990	0.3916	0.9074	0.9100	+1.85
Block 30 min	Humidity (%)	2.2422	2.2122	0.7353	0.7407	+1.34
Block 30 min	Pressure (hPa)	0.1769	0.1726	0.9891	0.9896	+2.43
Block 1 h	Temperature ($^{\circ}$ C)	0.7494	0.7411	0.7051	0.7095	+1.11
Block 1 h	Humidity (%)	3.6803	3.6478	0.4830	0.4910	+0.88
Block 1 h	Pressure (hPa)	0.2741	0.2711	0.9701	0.9705	+1.09
Block 6 h	Temperature ($^{\circ}$ C)	1.0237	1.0236	0.3515	0.3513	+0.01
Block 6 h	Humidity (%)	6.9979	6.9926	0.0041	0.0048	+0.08

Scenario	Parameter	MAE KNN	MAE Hyb	R ² KNN	R ² Hyb	Δ MAE (%)
Block 6 h	Pressure (hPa)	1.2254	1.2243	0.4848	0.4849	+0.09

Overall, the advantage of the hybrid approach is not absolute but depends on the structure and duration of the missing data. This finding complements earlier studies that generally report hybrid superiority in aggregate without mapping its boundary of validity (Luo et al., 2019; Liu et al., 2020). For operational systems the implication is clear: applying a static weight risks degrading performance under certain conditions, so adaptive weighting becomes an essential component rather than an optional refinement.

The magnitude of improvement observed here is smaller than that reported by several hybrid studies in other domains. Two factors plausibly explain the difference. First, those studies address forecasting, where the temporal model extrapolates beyond the observed record and its contribution is inherently larger, whereas the present task is interpolation within a gap flanked by valid observations, a setting in which a well-tuned seasonal analogue is already a strong baseline. Second, the meteorological parameters examined here exhibit pronounced diurnal regularity, so the climatological component of KNN captures much of the recoverable signal before the LSTM is consulted.

Because the hybrid never fell materially below the baseline on block gaps and converged to it rather than degrading on long gaps, the adaptive weighting can be regarded as a safeguard as much as an accuracy mechanism. In an operational setting where gap length is unknown in advance, this property is arguably more valuable than a larger average gain that carries the risk of severe degradation in unfavourable conditions.

The strength of the developed system lies in its ability to operate continuously with an explicit audit trail, since every data row stores the status and the reconstruction method detail. Its limitation is that the system was tested at a single station, so generalization to regions with different climatic characteristics cannot yet be assured, and the magnitude of the hybrid advantage is small, meaning its principal benefit lies in stability rather than a leap in accuracy.

Conclusion

A hybrid LSTM-KNN model with dynamic weighted blending was successfully designed and implemented in a web-based system using Streamlit and SQLite, covering layered quality control, missing data reconstruction, and a monitoring dashboard. Based on evaluation across nine artificial gap scenarios for three continuous parameters, the effectiveness of the model is conditional on the structure and duration of the missing data. On sporadic MCAR gaps the hybrid offers no

advantage, because isolated gaps do not reach the LSTM activation threshold. On block gaps the hybrid is consistently superior, with the highest MAE reduction of 2.66% at a 10-minute duration, shrinking to 0.07% at 6 hours as the weight decay takes effect. Convergence towards KNN-only performance at long gaps shows that the decay mechanism fulfils its function of preventing autoregressive drift. Future work is recommended to extend testing to gaps longer than six hours, to develop a two-stage approach for the rainfall parameter, and to test the model at several stations with different climatic characteristics.

Acknowledgements

The authors thank the State College of Meteorology Climatology and Geophysics for academic support, and the Meteorology, Climatology and Geophysical Agency for access to the AWS data used in this research.

References

- Afrifa-Yamoah, E., Mueller, U. A., Taylor, S. M., & Fisher, A. J. (2020). Missing data imputation of high-resolution temporal climate time series data. *Meteorological Applications*, 27(1). <https://doi.org/10.1002/met.1873>
- 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>
- Akkem, Y., Biswas, S. K., & Varanasi, A. (2024). Streamlit-based enhancing crop recommendation systems with advanced explainable artificial intelligence for smart farming. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-024-10208-z>
- Almeida, A., Bras, S., Sargento, S., & Pinto, F. C. (2024). Focalize K-NN: An imputation algorithm for time series datasets. *Pattern Analysis and Applications*, 27(2). <https://doi.org/10.1007/s10044-024-01262-3>
- Boomgard-Zagrodnik, J. P., & Brown, D. J. (2022). Machine learning imputation of missing Mesonet temperature observations. *Computers and Electronics in Agriculture*, 192, 106580. <https://doi.org/10.1016/j.compag.2021.106580>
- Cerqueira, V., & Torgo, L. (2023). Multi-output ensembles for multi-step forecasting

- (arXiv:2306.14563). arXiv. <https://arxiv.org/abs/2306.14563>
- Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022). RESTful API testing methodologies: Rationale, challenges, and solution directions. *Applied Sciences*, 12(9), 4369. <https://doi.org/10.3390/app12094369>
- Gaffney, K. P., Prammer, M., Brasfield, L., Hipp, D. R., Kennedy, D., & Patel, J. M. (2022). SQLite: Past, present, and future. *Proceedings of the VLDB Endowment*, 15(12), 3535-3547. <https://doi.org/10.14778/3554821.3554842>
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. <https://otexts.com/fpp3/>
- Laeq, A., Li, J., & Adeel, U. (2026). A hybrid LSTM-attention model for multivariate time series imputation: Evaluation on environmental datasets. *Machine Learning and Knowledge Extraction*, 8(1), 18. <https://doi.org/10.3390/make8010018>
- Lattifia, T., Buana, P. W., & Rusjyanthi, N. K. D. (2022). Model prediksi cuaca menggunakan metode LSTM. *JITTER: Jurnal Ilmiah Teknologi dan Komputer*, 3(1), 994-1002. <https://doi.org/10.24843/jtrti.2022.v03.i01.p35>
- Liu, M., Huang, Y., Li, Z., Tong, B., Liu, Z., Sun, M., Jiang, F., & Zhang, H. (2020). The applicability of LSTM-KNN model for real-time flood forecasting in different climate zones in China. *Water*, 12(2), 440. <https://doi.org/10.3390/w12020440>
- Luo, X., Li, D., Yang, Y., & Zhang, S. (2019). Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation*, 2019, 1-10. <https://doi.org/10.1155/2019/4145353>
- Nida, H., Kashif, M., Khan, M. I., & Ghamkhar, M. (2023). Comparison of missing data imputation methods using weather data. *Pakistan Journal of Agricultural Sciences*, 60(2), 327-336. <https://doi.org/10.21162/PAKJAS/23.228>
- Nugraha, Y. E. N., Ariawan, I., & Arifin, W. A. (2023). Weather forecast from time series data using LSTM algorithm. *Jurnal Teknologi Informasi dan Komunikasi*, 14(1), 144-152. <https://doi.org/10.51903/jtikp.v14i1.531>
- Nugroho, A. A., & Haris, M. (2024). Analisis efektivitas teknik imputasi pada LSTM untuk meningkatkan kualitas data pada peramalan curah hujan. *Jurnal Informatika dan Rekayasa Elektronik*, 7(2), 301-311. <https://doi.org/10.36595/jire.v7i2.1172>
- Oktaviani, I., & Putrada, A. (2022). KNN imputation to missing values of regression-based rain duration prediction on BMKG data. *Jurnal Infotel*, 14(4), 249-254. <https://doi.org/10.20895/infotel.v14i4.840>
- Ouyang, Z., Ravier, P., & Jabloun, M. (2022). Are deep learning models practically good as promised? A strategic comparison of deep learning models for time series forecasting. In *Proceedings of the 2022 European Signal Processing Conference (EUSIPCO)* (pp. 1477-1481). <https://eurasip.org/Proceedings/Eusipco/Eusipco2022/pdfs/0001477.pdf>
- Purwandari, K., Aufauzan, B. T., & Sigalingging, J. W. C. (2026). Web-based quality control dashboard design for data validation and monitoring: A case study of BMKG instruments. *Engineering, Mathematics and Computer Science Journal*, 8(1).
- Sugiarto, B., Rahmat, R. F., & Purnamawati, S. (2019). Rancang bangun Automatic Weather Station berbasis mikrokontroler. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 6(3), 245-252.
- Teutsch, P., & Mäder, P. (2022). Flipped classroom: Effective teaching for time series forecasting. *Transactions on Machine Learning Research*. <https://openreview.net/forum?id=w3x20YEcQK>
- Wahyudi, A., & Widyarto, W. O. (2025). Deteksi anomali dan rekonstruksi data pengamatan otomatis berbasis machine learning. *Jurnal Instrumentasi dan Teknologi Meteorologi*, 8(1), 33-44.
- World Meteorological Organization. (2023). *Guide to instruments and methods of observation* (WMO-No. 8, Vol. V).
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295-316. <https://doi.org/10.1016/j.neucom.2020.07.061>