



LSTM-Based Soft Sensor for Anomaly Detection in AWS Temperature and Humidity Data

Muhamad Dzakwan Firdaus^{1*}, Muchammad Rizqy Nugraha², Maulana Putra², Marzuki Sinambela², Dibyo Susanto²

¹ Instrumentation Study Program, College of Meteorology, Climatology, and Geophysics (STMKG), Tangerang City, Indonesia.

² College of Meteorology, Climatology, and Geophysics (STMKG), Tangerang City, Indonesia.

DOI: <https://doi.org/10.29303/goescienceed.v7i4.3311>

Article Info:

Received : August, 26 2026
Revised : September, 09 2026
Accepted : September, 29 2026
Published : October, 09 2026

Correspondence:

Muhamad Dzakwan Firdaus

Phone:

Abstract: Temperature and relative humidity records from an Automatic Weather Station (AWS) may contain missing values, abrupt deviations, or gradual shifts that are difficult to judge from a single observation. This study develops an LSTM-based soft-sensor that estimates the next temperature and humidity values from recent AWS observations. The difference between the observed and estimated values is treated as a residual. A P95 threshold is used for spike detection, while Two-Sided CUSUM is evaluated separately for gradual drift. The model was deployed as TensorFlow Lite on a Raspberry Pi 5 and evaluated using operational data from BMKG Soekarno-Hatta. On the test set, R^2 reached 0.9944 for temperature and 0.9705 for humidity. During the 13-day operational evaluation with 1,842 stored records, P95-based detection produced an F1-score of 96.29% and FAR of 1.83% for temperature, and an F1-score of 92.29% and FAR of 4.79% for humidity. CUSUM detected drift in both evaluation windows, but its false-alarm rate and detection delay changed markedly between periods. The results support residual-based monitoring on edge hardware while showing that CUSUM still needs further calibration before use as a primary operational alarm.

Keywords: Anomaly Detection; Soft-Sensor; LSTM; CUSUM; Automatic Weather Station

Citation: Firdaus, M. D., Nugraha, M. R., Putra, M., Sinambela, M., & Susanto, D. (2026). LSTM-Based Soft Sensor for Anomaly Detection in AWS Temperature and Humidity Data. *Jurnal Pendidikan, Sains, Geologi, Dan Geofisika (GeoScienceEd Journal)*, 7(4), 6122–6128. <https://doi.org/10.29303/goescienceed.v7i4.3311>

Introduction

Automatic Weather Station (AWS) observations are used repeatedly in weather monitoring, so data-quality problems can carry into later analysis. In a parallel evaluation at Kemayoran, Wicaksana & Putra (2021) reported high correlations between AWS and conventional instruments, but also mean differences of 2.3 °C for temperature and 11.3% for relative humidity, together with periods of reduced data availability. A systematic mapping study of IoT environmental monitoring likewise identifies calibration, data interpolation, and outlier detection as recurring data-quality measures and distinguishes outliers from missing values (Buelvas, Múnera, Tobón V, Aguirre, & Gaviria, 2023). These findings motivate an additional

check on whether an incoming AWS reading remains consistent with patterns learned from historical observations.

This study focuses on air temperature and relative humidity. Wicaksana & Putra (2021) show that both variables are measured within the AWS thermohygrometer system and that their differences from conventional observations are not identical. Santoso et al. (2023) also trained separate LSTM models for atmospheric temperature and relative humidity and reported different error-detection performance, with the relative-humidity model performing below the temperature model in their overall accuracy summary.

Previous studies have examined different approaches to sensor anomaly detection. Santoso et al.

(2023) developed separate LSTM models for AWS sensor-error detection, while Muñoz-Molina, Cazorla-Piñar, Dominguez-Morales, Lafuente, & Perez-Peña (2022) compared neural-network estimates with sensor measurements to identify uncalibration. Noh & Moon (2023) investigated LSTM-based anomaly detection in indoor environmental monitoring, and D'Agostino, Violante, & MacArio (2024) studied temperature prediction on embedded hardware.

These studies address related problems, but provide limited evidence on the combined use of a multi-output LSTM soft sensor, residual-based spike and drift detection, and edge deployment for AWS monitoring. Two questions remain relevant to this approach: how detection performance differs between temperature and relative humidity, and how consistently CUSUM detects gradual shifts across different evaluation periods without producing excessive false alarms or detection delays.

Large residuals in a single cycle are handled separately from smaller deviations that persist over time. For spikes, this study uses the 95th percentile of validation residuals as the threshold. Komadina, Martinić, Groš, & Mihajlović (2024) describe the percentile method as setting a threshold at the k -th percentile of anomaly scores, which provides the methodological basis for this choice. Drift is evaluated with Two-Sided CUSUM because CUSUM accumulates deviations over time and is sensitive to smaller shifts (Riaz, Alshammari, Abbas, & Mahmood, 2025). Its behavior depends strongly on the chosen parameters, and Lopes, Barbon Junior, & de Melo (2025) show that a single hyperparameter configuration may not remain suitable across an entire time series. For that reason, the CUSUM result is reported separately from the P95 spike confusion matrix.

This study develops an AWS monitoring system that combines a multi-output LSTM soft sensor with two residual-based anomaly detection methods. The LSTM estimates air temperature and relative humidity one step ahead. The residuals are then evaluated separately using P95 thresholding for spike detection and Two-Sided CUSUM for gradual drift detection. The model is converted to TensorFlow Lite and deployed on a Raspberry Pi 5, which retrieves observations through the BMKG API, stores the detection results, and updates a web dashboard.

The study evaluates prediction accuracy using a chronological test set and assesses spike detection through controlled fault injections. CUSUM performance is examined separately using historical holdout and post-training operational data to measure event detection rates, false alarms, and detection delays. The main contribution is the implementation and evaluation of these methods within a single edge-based

monitoring system using data from BMKG Soekarno-Hatta AWS.

Method

Data and Dataset Split

Historical data were obtained from the BMKG Soekarno-Hatta AWS at an interval of approximately 10 minutes. Four base variables were used as inputs: mean air temperature, 10-minute minimum temperature, prevailing wind direction, and relative humidity. The data were split chronologically into 70% training, 15% validation, and 15% testing. The validation set was used to determine the P95 threshold and the CUSUM reference standard deviation, whereas the test set was reserved for MAE, RMSE, R^2 , MAPE, and BIAS. Separate training, validation, and test partitions are also used in sensor-monitoring studies by Muñoz-Molina et al. (2022) and Santoso et al. (2023), the 70/15/15 chronological split used here is specific to this study. Keeping the sets separate prevents the final evaluation records from determining the thresholds being evaluated.

Missing values were handled according to gap length: linear interpolation was used for gaps of up to two samples, forward filling for gaps of three to six samples, and longer gaps were removed. These cutoffs are implementation choices in this study, not values taken from the cited literature. Buelvas et al. (2023) identify interpolation and outlier handling as common data-quality techniques in sensor monitoring and link interpolation to data completeness. Min-max normalization parameters were fitted only on the training set and then applied unchanged to validation and test data. The chronological order was preserved because the model uses the previous 100 observations as its lookback window.

LSTM Soft-Sensor Architecture

The model contains three LSTM layers with 128, 128, and 64 units, each followed by 0.2 dropout. Dense layers with 32 and 16 units feed two linear outputs for temperature and relative humidity. Training used Adam with an initial learning rate of 0.001, a batch size of 512, and mean squared error as the loss function; ReduceLROnPlateau and early stopping controlled the training schedule. These layer sizes and training settings are specific to this study. Their use in a sequential soft-sensor model is consistent with published LSTM soft-sensor work that models temporal dependencies in process data (Liu, Ni, & Wang, 2024) and with LSTM-based environmental anomaly models built from time-series measurements (Noh & Moon, 2023).

Table 1. LSTM hyperparameter configuration

Hyperparameter	Value
LSTM architecture	128 - 128 - 64 unit, Dropout 0.2 each layer
Dense Layer	32 - 16 - 2
Maximum Epoch	100
Initial learning rate	0,001 (Adam)
Learning-rate schedule	ReduceLROnPlateau, factor 0,5 patience 4
Early stopping	Patience 12 epoch, restore best weights
Lookback	100 time steps
Batch size	512
Loss Function	Mean Squared Error

Residual-Based Detection

At each cycle, this study defines the signed residual as the observed value minus the soft-sensor estimate. The absolute residual is used for spike detection, while the signed residual is retained for CUSUM. Comparing a model-derived estimate with the measured sensor behavior is an established sensor-monitoring strategy: Muñoz-Molina et al. (2022) compare inferred and measured sensor values to identify uncalibration, while Maleki, Maleki, & Jennings (2021) use model reconstruction error as the quantity that is thresholded for anomaly detection. The residual definition used here applies that general error-based idea to the LSTM soft-sensor output.

$$s_t = |y_t - \hat{y}_t|$$

The spike threshold is based on the 95th percentile of the absolute residuals in the validation set and is not recalculated on operational data. Komadina et al. (2024) define the percentile method as selecting the k -th percentile of anomaly scores as the threshold; the P95 choice in this study follows that construction. The final inference configuration uses a temperature threshold of 0.927985 °C. For relative humidity, the base P95 value of 3.054376 was multiplied by a calibration factor of 1.25, producing an effective threshold of 3.817970 percentage points. The factor 1.25 is specific to this validation set: it was the smallest tested candidate that reduced the validation false alarm rate to no more than 2%.

$$\text{Threshold}_p = \text{Percentile}(|\text{residual}_i|, p), i = 1, 2, \dots, n$$

CUSUM uses two nonnegative accumulators to track positive and negative shifts. Riaz et al. (2025)

describe CUSUM in terms of a reference value k and a decision limit h and note that the method is more sensitive to smaller process shifts and drifts than a memoryless chart. In this implementation, the residual is centered on an adaptive median baseline, the allowance is $k\sigma$ with $k = 0.5$, and σ is obtained from validation residuals. The adaptive median and the 1.5h accumulator cap are implementation choices in this study rather than constants taken from Riaz et al. (2025). Lopes et al. (2025) further show that CUSUM effectiveness depends strongly on hyperparameter configuration, which is why the selected settings are evaluated rather than assumed to generalize across periods.

$$d_t = e_t - \mu_t$$

$$C_t^+ = \max(0, C_{t-1}^+ + (x_t - \mu) - k \cdot \sigma)$$

$$C_t^- = \max(0, C_{t-1}^- - (x_t - \mu) - k \cdot \sigma)$$

Table 2. Parameter P95 and CUSUM parameters in the final configuration

Parameter	Temperature	Relative humidity
P95 spike threshold	0,927985	3,817970
Reference σ	0,537562	2,030948
k	0,5	0,5
Allowance $k\sigma$	0,268781	1,015474
CUSUM threshold factor	$4,0 \times k$	$2,5 \times k$
h	2,150248	5,077370
$C_{\max} = 1,5h$	3,225372	7,616055

Implementation and Evaluation

The Raspberry Pi 5 used in this study has a 4 GB of RAM. Services run through Docker Compose. A Python container retrieves BMKG API data and runs TensorFlow Lite inference, MySQL stores the records, and the web service updates the dashboard. D'Agostino et al. (2024) demonstrate containerized LSTM temperature processing on Raspberry Pi hardware, and Javed et al. (2024) discuss machine-learning deployment under constrained IoT resources. In the present system, converting the Keras model reduced the file size from about 1.45 MiB to 264.33 KiB. Keras and TensorFlow Lite latency were not compared, so the measured benefit reported here is the smaller model file, not a claimed inference speedup.

Operational testing ran from June 28 to July 10, 2026. At a 10-minute interval, 1,872 records would have been expected; 1,842 were stored successfully, while 30 cycles had no usable API record. During this period, 52 controlled fault-injection events were scheduled.

Temperature offsets ranged from 1.5 to 4.0 °C, and relative-humidity offsets ranged from 6.2 to 14.6 percentage points. The injection schedule served as the ground truth for spike evaluation, while CUSUM was evaluated separately through replay on one holdout window and one live post-training window. Controlled synthetic errors are also used in AWS sensor-error research to create known abnormal conditions (Santoso et al., 2023), although the offset magnitudes, timing, and evaluation protocol reported here are specific to this study.

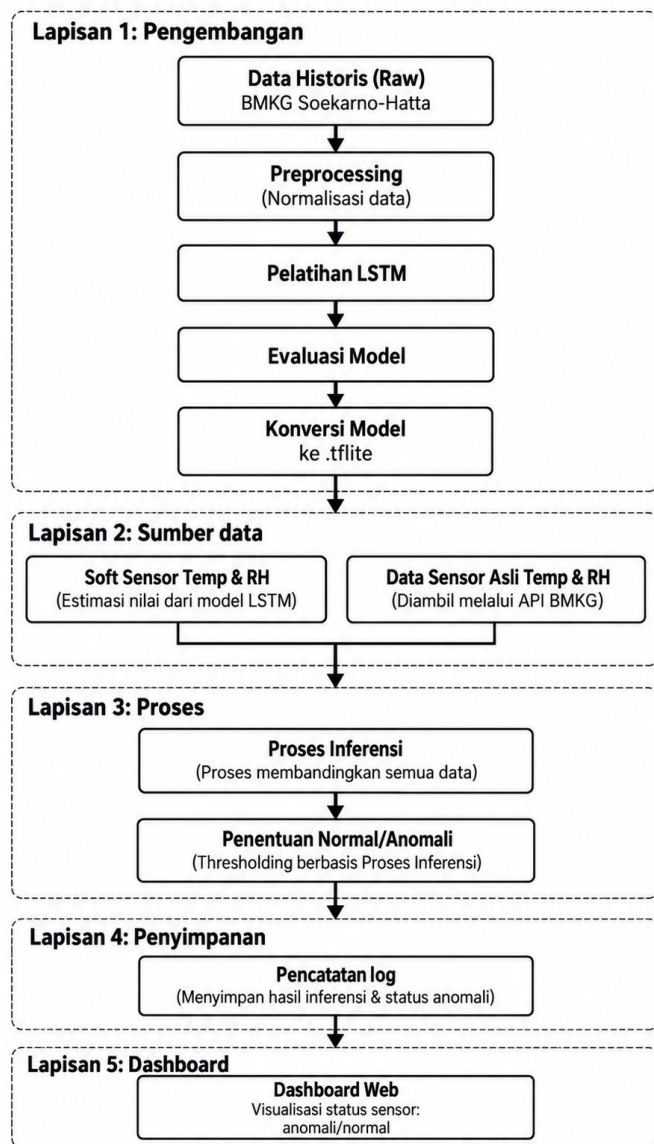


Figure 1. Research workflow from model development to deployment on the Raspberry Pi 5

Result and Discussion

Soft-Sensor Performance

On the chronological test set, R^2 reached 0.9944 for temperature and 0.9705 for relative humidity. Temperature MAE was 0.1593 °C and relative-humidity

MAE was 1.4958 percentage points. Liu et al. (2024) use MAE, RMSE, and R^2 to assess soft-sensor estimation, so the same metrics are used here to describe estimation error and trend agreement. Temperature BIAS was 0.0395 °C. For relative humidity, BIAS was -1.1324; because this study defines BIAS as observed minus estimated, the negative sign means the estimates tended to lie above the observations. That shift is relevant when reading the detector results, but it does not by itself establish the cause of every false positive.

Table 3. Soft-sensor performance on the test set

Metrik	Suhu (°C)	Kelembaban (%)
MAE	0,1593	1,4958
RMSE	0,2548	1,7700
R^2	0,9944	0,9705
MAPE	0,5463%	1,7539%
BIAS	0,0395	-1,1324

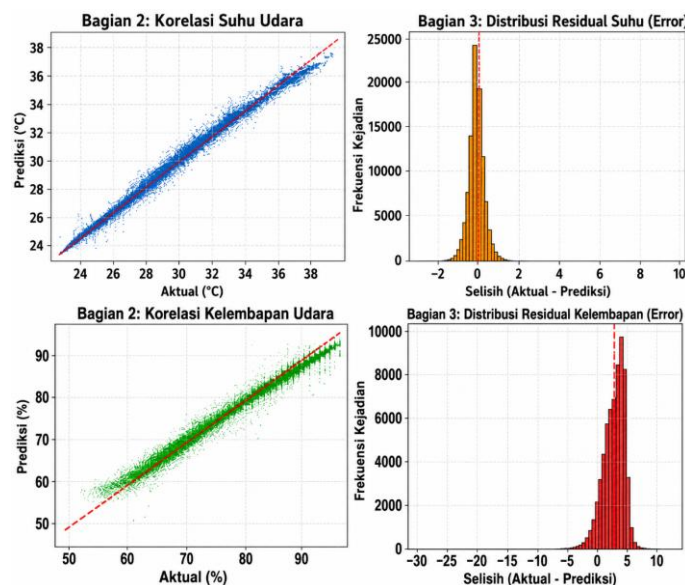


Figure 2. Observed-versus-estimated relationship and residual distributions for temperature and relative humidity

The P95 thresholds derived from the validation set were used without adjustment during the operational period. Temperature produced 415 true positives, 1,395 true negatives, 26 false positives, and 6 false negatives, giving an F1-score of 96.29% and a false alarm rate of 1.83%. Relative humidity produced 419 true positives, 1,353 true negatives, 68 false positives, and 2 false negatives; recall reached 99.52%, precision was 86.04%, and the false alarm rate was 4.79%. Precision, recall, and F1-score are also used to evaluate LSTM-based environmental anomaly detection by Noh & Moon (2023), while Komadina et al. (2024) show that

threshold selection is a distinct part of anomaly-detector design. The figures here are therefore reported as results of this dataset and threshold configuration, not as direct performance comparisons with those studies.

Table 4. Results of P95-Based Spike Detection Evaluation

Metric	Temperature	Relative humidity
TP / TN / FP / FN	415 / 1395 / 26 / 6	419 / 1353 / 68 / 2
Precision	94,10%	86,04%
Recall	98,57%	99,52%
F1-Score	96,29%	92,29%
Accuracy	98,26%	96,20%
FAR	1,83%	4,79%

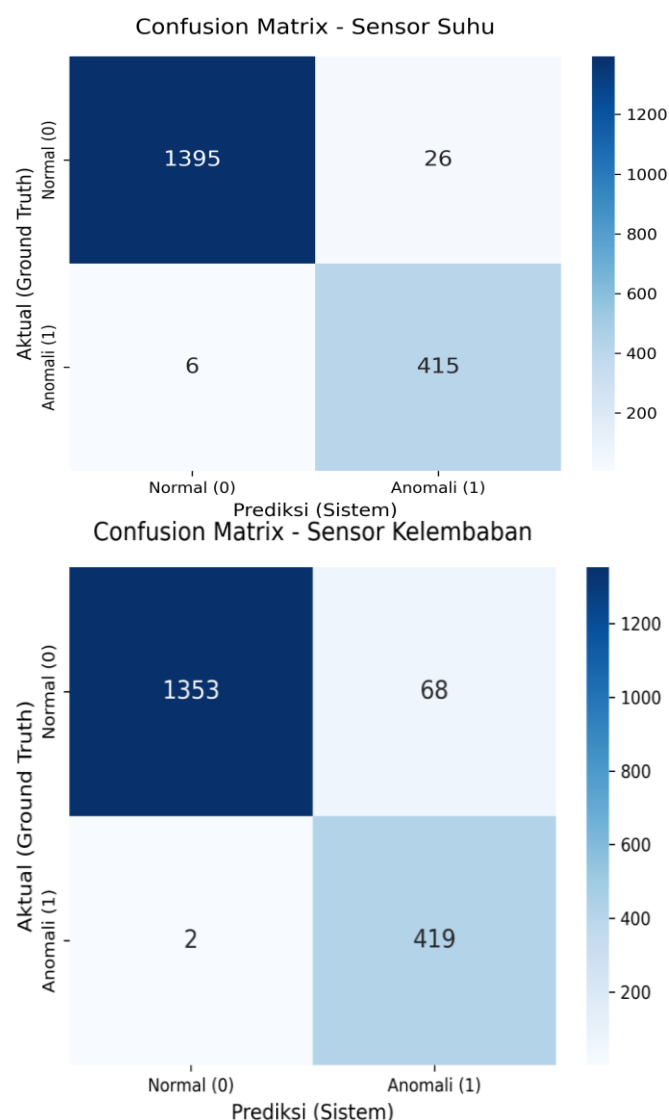


Figure 3. Confusion matrices for temperature and relative-humidity spike detection

The two channels show different operating trade-offs. Relative humidity missed almost none of the injected anomalies, but it marked more normal records as anomalous. Maleki et al. (2021) note that anomaly detectors can face a trade-off between detection performance and false-alarm behavior and report that thresholded LSTM-based methods can produce both false positives and false negatives across datasets. In this study, the humidity residual bias is one factor that should be considered when interpreting the higher false alarm rate, but the available data do not justify assigning all false positives to that bias. The P95 threshold was derived from validation residuals and was not retuned after the operational results were observed.

CUSUM Drift Evaluation

USUM was tested on two time windows using the same configuration. Window A, a January 2026 holdout period, produced an event detection rate (EDR) of 100% for both temperature and relative humidity. Mean detection delay was about 10.8 minutes for temperature and 14.4 minutes for humidity, but the sample-level false alarm rates were 55.86% and 45.43%, respectively. In Window B, which used live data from June-July 2026, the false alarm rate fell to 9.84% for temperature and 11.65% for humidity, while detection delay increased to about three hours and EDR declined to 93.33% and 86.67%. Riaz et al. (2025) describe CUSUM as a memory chart designed to accumulate departures from a target and improve sensitivity to small shifts; the two-window test here examines how that behavior transfers to the residual series used in this study.

Table 5. Results of CUSUM

Window	Channel	Events	EDR	Detection delay	FAR/sample
A holdout	Temp	16	100,00%	10,8 menit	55,86%
A holdout	RH	16	100,00%	14,4 menit	45,43%
B - live	Temp	15	93,33%	189,3 menit	9,84%
B - live	RH	15	86,67%	179,3 menit	11,65%

CUSUM detected drift in both periods, but the same configuration did not behave consistently across them. Riaz et al. (2025) show that CUSUM response depends on the reference value and decision limit as well as the monitored process. Lopes et al. (2025) make the same issue explicit for time series, reporting that a single CUSUM hyperparameter setup may not remain

suitable across an entire series and may require retuning. The present result follows that pattern: Window A reacted quickly but generated many sample-level false alarms, whereas Window B reduced false alarms at the cost of longer detection delays and missed events. CUSUM is therefore treated as a separate drift-analysis channel rather than the primary operational alarm in this study.

Edge Deployment and Limitations

The Raspberry Pi 5 handled data retrieval, LSTM inference, residual calculation, anomaly detection, storage, and dashboard updates in the deployed system. One monitoring snapshot showed about 600.8 MB of memory used by the Python container and CPU usage below 1%. D'Agostino et al. (2024) likewise report CPU and memory measurements for containerized LSTM processing on Raspberry Pi hardware, illustrating why resource use is relevant to embedded deployment. Javed et al. (2024) also frame computational limits as a central constraint for machine learning on IoT devices. The values reported here come from a single observation rather than a continuous 13-day log, so they show that the system ran on the selected hardware but do not establish average resource use or computational optimality.

The evaluation is limited to one station and a 13-day operational period, and most ground-truth anomaly labels came from controlled fault injections. Santoso et al. (2023) also use synthetic AWS sensor-error scenarios, but their LSTM models are trained to recognize normal and synthetic-error conditions directly. Here, the LSTM only provides a reference estimate and the anomaly decision is made from the residual, so a direct performance comparison would mix different datasets and evaluation protocols. Javed et al. (2024) and Imran et al. (2023) are retained only as examples of anomaly-detection evaluation on IoT or resource-constrained systems; their intrusion-detection tasks are not meteorological sensor validation. These differences limit how far the present results can be generalized beyond the tested station, period, and injected disturbance patterns.

Conclusion

The LSTM soft sensor ran on a Raspberry Pi 5 and produced reference estimates for AWS temperature and relative humidity. On the test set, R^2 reached 0.9944 for temperature and 0.9705 for relative humidity. Across 1,842 operational records, P95-based spike detection produced an F1-score of 96.29% with a 1.83% false alarm rate for temperature and an F1-score of 92.29% with a 4.79% false alarm rate for relative humidity. The use of model-derived estimates as a reference for measured sensor behavior is consistent with the estimate-versus-

measurement approach of Muñoz-Molina et al. (2022), while Liu et al. (2024) show how LSTM-based soft sensors can represent temporal process variables and be evaluated against actual values. The numerical results in this paragraph are specific to the present dataset and evaluation protocol.

CUSUM detected drift in both evaluation windows, but its performance was not stable across periods. The holdout window produced fast detection with a high false alarm rate, whereas the live window reduced false alarms but extended mean detection delay to roughly three hours. This sensitivity to configuration is consistent with Riaz et al. (2025), who describe the role of CUSUM reference and decision parameters, and with Lopes et al. (2025), who show that one hyperparameter configuration may not suit an entire time series. The next step is therefore to recalibrate CUSUM and compare it with alternative drift detectors under the same data and disturbance scenarios. Broader testing should also include additional AWS sites, examine the relative-humidity bias, and record inference latency and resource use continuously.

Acknowledgements

The authors would like to express their sincere gratitude to the College of Meteorology, Climatology, and Geophysics (STMKG) for its support and contributions to this research.

References

- Buelvas, J., Múnera, D., Tobón V, D. P., Aguirre, J., & Gaviria, N. (2023, April 1). Data Quality in IoT-Based Air Quality Monitoring Systems: a Systematic Mapping Study. *Water, Air, and Soil Pollution*, Vol. 234. Institute for Ionics. <https://doi.org/10.1007/s11270-023-06127-9>
- D'Agostino, P., Violante, M., & MacArio, G. (2024). Optimizing Lstm-Based Temperature Prediction Algorithm for Embedded System Deployment. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ETFA61755.2024.10711142>
- Imran, Zuhairi, M. F. A., Ali, S. M., Shahid, Z., Alam, M. M., & Su'ud, M. M. (2023). Improving Reliability for Detecting Anomalies in the MQTT Network by Applying Correlation Analysis for Feature Selection Using Machine Learning Techniques. *Applied Sciences (Switzerland)*, 13(11). <https://doi.org/10.3390/app13116753>

- Javed, A., Ehtsham, A., Jawad, M., Awais, M. N., Qureshi, A. U. H., & Larijani, H. (2024). Implementation of Lightweight Machine Learning-Based Intrusion Detection System on IoT Devices of Smart Homes. *Future Internet*, 16(6). <https://doi.org/10.3390/fi16060200>
- Komadina, A., Martinić, M., Groš, S., & Mihajlović, Ž. (2024). Comparing Threshold Selection Methods for Network Anomaly Detection. *IEEE Access*, 12, 124943–124973. <https://doi.org/10.1109/ACCESS.2024.3452168>
- Liu, Y., Ni, D., & Wang, Z. (2024). A Fault-Tolerant Soft Sensor Algorithm Based on Long Short-Term Memory Network for Uneven Batch Process. *Processes*, 12(3). <https://doi.org/10.3390/pr12030495>
- Lopes, J. F., Barbon Junior, S., & de Melo, L. F. (2025). Online Meta-Recommendation of CUSUM Hyperparameters for Enhanced Drift Detection. *Sensors*, 25(9). <https://doi.org/10.3390/s25092787>
- Maleki, S., Maleki, S., & Jennings, N. R. (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. *Applied Soft Computing*, 108. <https://doi.org/10.1016/j.asoc.2021.107443>
- Muñoz-Molina, L. J., Cazorla-Piñar, I., Dominguez-Morales, J. P., Lafuente, L., & Perez-Peña, F. (2022). Real-time detection of uncalibrated sensors using neural networks. *Neural Computing and Applications*, 34(10), 8227–8239. <https://doi.org/10.1007/s00521-021-06865-z>
- Noh, S. H., & Moon, H. J. (2023). Anomaly Detection Based on LSTM Learning in IoT-Based Dormitory for Indoor Environment Control. *Buildings*, 13(11). <https://doi.org/10.3390/buildings13112886>
- Riaz, M., Alshammari, H., Abbas, N., & Mahmood, T. (2025). Navigating Process Drift: The Power of CUSUM in Monitoring Air Quality Processes and Maintenance Operations. *Arabian Journal for Science and Engineering*, 50(14), 11145–11166. <https://doi.org/10.1007/s13369-024-09453-0>
- Santoso, B., Ryan, M., Subyantara Wicaksana, H., Ananda, N., Budiawan, I., Mukhlis, F., & Kurniadi, D. (2023). Predictive Maintenance: Automatic Weather Station Sensors Error Detection using Long Short-Term Memory. *Ultima Computing : Jurnal Sistem Komputer*, 15(2), 41.
- Wicaksana, H. S., & Putra, M. (2021). Evaluasi Kinerja Automatic Weather Station Berdasarkan Pengamatan Paralel di Stasiun Meteorologi Kemayoran. *Seminar Nasional Teknik Elektro*, 6, 59–64. Retrieved from <https://prosiding.pnj.ac.id/index.php/SNTE/article/view/910>