Vol. 12 No.1 Tahun 2026, pp 33-43
Jurnal Teknologi dan Manajemen Informatika
http:// http://jurnal.unmer.ac.id/index.php/jtmi
P-ISSN: 1693-6604 E-ISSN: 2580-8044

Comparative Analysis of Unsupervised Methods for Anomaly Detection in IoT-Based Pharmaceutical Cold Chain Temperature

Yusof Zaky[1], Alva Hendi Muhammad[2]

Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Indonesia

Article Info

Article History
Received : 22-04-2026
Revised : 05-06-2026
Accepted : 10-06-2026

Keywords
Anomaly Detection;
Cold Chain Monitoring;
Internet of Things;
LSTM;
Time Series;

Corresponding Author
Yusof Zaky,
Universitas Amikom Yogyakarta,
zakyyusof@gmail.com

ABSTRACT

Maintaining vaccines within the 2°C–8°C range throughout cold chain storage and distribution is essential, since temperature excursions can degrade their biological potency. The growth of IoT-enabled sensors now allows continuous temperature data collection, opening the door to automated anomaly detection via time-series analysis. This research compares several unsupervised approaches for spotting temperature anomalies in an IoT-based pharmaceutical cold chain setup, benchmarking Isolation Forest against three deep learning architectures: Autoencoder, LSTM, and LSTM-Attention. Using roughly 8,640 temperature readings collected at 5-minute intervals over 30 days, the data were normalized with Min-Max Scaling and structured into sequences via a sliding window technique. Performance was assessed using precision, recall, and F1-score, alongside MAE and RMSE for prediction accuracy. Results showed Isolation Forest outperforming the other models (precision: 0.686, recall: 0.418, F1-score: 0.520) while also being the fastest to train and run. The deep learning models underperformed, likely limited by dataset size, making Isolation Forest the more practical choice for balancing detection accuracy with computational cost.

INTRODUCTION

Vaccines are biological products that are highly sensitive to temperature fluctuations, requiring a stable storage system during distribution and storage processes. According to the World Health Organization (WHO), most vaccines must be stored within a temperature range of 2°C to 8°C to maintain their stability and effectiveness [1]. Exposure to temperatures outside this range may reduce vaccine potency and compromise immunization programs [2]. Progress in IoT technology has made it possible to conduct uninterrupted temperature surveillance using sensors linked to communication infrastructure, enabling ongoing data acquisition [2], [3]. IoT sensors have been widely applied in logistics systems to monitor environmental conditions during the transportation and storage of temperature-sensitive products [3].

The problem of identifying anomalies within time-series data has been a subject of widespread investigation across numerous fields. According to Chandola et al. [4], anomalies refer to observations that diverge from anticipated patterns, and they are broadly categorized into three types: point anomalies, contextual anomalies, and collective anomalies. In time-series sensor data analysis, deep learning methods such as Autoencoder and Long Short-Term Memory (LSTM) have been widely used for anomaly detection [5], [6]. Recent studies indicate that more advanced architectures, including transformer-based models and hybrid approaches, can further improve anomaly detection performance [7], [8]. Autoencoders operate by

constructing compact encodings of normal patterns and subsequently identifying irregularities through elevated reconstruction error values [9], while LSTM is designed to capture long-term dependencies in sequential data [5]. By dynamically weighting individual timesteps according to their relevance, the attention mechanism contributes to improved model sensitivity and predictive accuracy [10], [11].

On the other hand, Isolation Forest employs an ensemble strategy in which anomalous instances are separated from normal data through the use of randomly constructed decision trees [20]. This method does not rely on data distribution assumptions and does not require learning complex temporal patterns, making it more robust for small datasets. Several studies have developed IoT-based cold chain monitoring systems for real-time vaccine storage monitoring [7], [3]. However, most studies rely on a single anomaly detection model without performing a comparative analysis of multiple methods. Ren et al. [4] conducted a large-scale evaluation of anomaly detection services, but their study focused on IT infrastructure rather than pharmaceutical cold chain systems.

Most existing studies use large-scale datasets and complex architectures, which are not suitable for IoT systems with limited data and computational resources. To the best of the authors' knowledge, no prior study has systematically compared classical machine learning and deep learning-based anomaly detection methods specifically under the constraints of a small-scale pharmaceutical cold chain IoT dataset, where both data scarcity and class imbalance are simultaneously present. Therefore, this study performs a comparative analysis of four anomaly detection methods: Isolation Forest, Autoencoder, LSTM, and LSTM-Attention. The main contributions of this study are: (1) comparing classical machine learning and deep learning methods for anomaly detection in small-scale time-series data;(2) evaluating computational efficiency in terms of training and inference time; and (3) analyzing the impact of limited data and class imbalance on model performance in IoT environments [25].

METHODS

This study adopts an experimental approach by comparing four anomaly detection methods applied to time-series temperature data collected from an IoT-based cold chain monitoring system. The methods consist of one baseline model, Isolation Forest, and three deep learning models: Autoencoder (AE), LSTM, and LSTM with an Attention mechanism (LSTM-Attention).

Dataset

The dataset used in this study consists of time-series temperature data obtained from an IoT-based cold chain monitoring system. The data were recorded at 5-minute intervals over a period of 30 days, resulting in approximately 8,640 data points. Each record includes a timestamp, temperature value, event type, and anomaly label. Although the temperature data were collected from a real IoT-based monitoring system, naturally occurring anomalies are extremely rare in well-maintained cold chain environments, making it practically infeasible to obtain sufficient label anomaly samples from real operational data alone. Anomaly simulation was therefore applied to inject controlled disturbances that represent realistic cold chain failure scenarios, enabling rigorous evaluation against a known ground truth. This approach is consistent with established practice in IoT anomaly detection research [23], [24], [25]. The dataset was divided into training and testing sets using a sequential split to preserve temporal order. Approximately 80% of the data (±6,912 samples) were used for training to learn normal temperature patterns, while the remaining 20% (±1,728 samples) were used for testing to evaluate anomaly detection performance.

Data Preprocessing

Several preprocessing steps were applied before training the models:

- Data Cleaning: invalid data were removed, including null values, duplicate timestamps, and physically impossible sensor readings (e.g., temperatures below −20°C or above 50°C).
- Normalization: Temperature values were normalized using Min-Max Scaling to transform the data into the range [0, 1], improving numerical stability during model training:
  $x\_norm = (x - x\_min)/(x\_max - x\_min)$ (1)
- Sliding Window: A sliding window technique was applied to transform the time-series data into sequences. Each input sample consists of 20 consecutive temperature readings (window size = 20 timesteps, equivalent to 100 minutes), enabling the models to capture temporal dependencies.

Anomaly Simulation

The dataset consists of normal temperature data combined with simulated anomaly events to represent real-world disturbances in cold chain systems. Three types of anomalies were introduced:

- Point Anomaly - Sudden temperature spikes between 5–10°C lasting for 1–3 timestamps (5–15 minutes), simulating short disturbances such as refrigerator door openings.
- Contextual Anomaly - Gradual temperature increases above 8°C lasting for 30–60 minutes, representing compressor performance degradation.
- Collective Anomaly - Sustained temperature increases over several hours, simulating system failures such as compressor malfunction or power outages.

Anomalies were injected by modifying temperature values outside the normal range (>8°C or <2°C). Data points within this abnormal range were labeled as 1 (anomaly), while normal data were labeled as 0.

Table 1. Class Distribution and Dataset

Class	Number of Data	Percentage
Normal (label = 0)	±8.208	±95%
Anomaly (label = 1)	±432	±5%
Total	**±8.640**	**100%**

Deep Learning Models

- Autoencoder (AE)

  An Autoencoder is a type of neural network that encodes input data into a lower-dimensional representation and subsequently reconstructs it, operating through an encoder-decoder structure [19]. This method has been proven effective for anomaly detection in sensor data because a model trained only on normal data will produce a high reconstruction error when receiving anomalous input. The encoder is responsible for extracting low-dimensional feature representations (latent representation) from input data, while the decoder reconstructs the input data from that representation. Anomaly detection is performed based on the reconstruction error value (Mean Absolute Error between input and model output). During training, the model is only exposed to normal data so that it learns to reconstruct normal patterns well. Data with reconstruction error values higher than a defined threshold are categorized as anomalies, because the model has difficulty reconstructing patterns it has never seen during training. The threshold value is determined using a statistical approach on the training data reconstruction errors: threshold = $\mu + k \cdot \sigma$, where μ is the mean reconstruction error computed over all training samples, σ is the corresponding standard deviation, and k is a sensitivity multiplier set to 3 (three-sigma rule). This approach assumes that reconstruction errors on normal data follow an approximately Gaussian distribution, and any data point exceeding the threshold is classified as an anomaly. The same threshold formula is applied consistently across Autoencoder and LSTM models to ensure a fair comparison.

$$threshold = mean(error\_train) + 3 \times \text{std}(error\_train)$$

The Autoencoder architecture used in this study is as follows:
Encoder: Dense(32, ReLU) → Dense(16, ReLU) → Dense(8, ReLU) (bottleneck).
Decoder: Dense(16, ReLU) → Dense(32, ReLU) → Dense(window_size, Linear).

- LSTM
  LSTM is a specialized form of Recurrent Neural Network that addresses the vanishing gradient limitation while retaining the ability to model extended dependencies across sequential inputs [5]. LSTM uses a gating mechanism (input gate, forget gate, output gate) to regulate information flow through the network. Malhotra et al. [22] show that LSTM can be effectively used for anomaly detection in multivariate time-series data using an error-based prediction approach. In this study, the LSTM model is used as a predictive model: given a sequence of temperature values over W timesteps, the model predicts the temperature value at the next timestep. Anomalies are detected based on prediction error between actual and predicted values, using the same threshold as Autoencoder.
  The LSTM architecture used is:
  LSTM (64 units, return_sequences = False) → Dense (1, Linear).

- LSTM with Attention Mechanism
  LSTM-Attention builds upon the standard LSTM framework by integrating an attention layer that dynamically allocates varying degrees of importance to individual timesteps within the input sequence [7][18]. The attention mechanism adopted in this study is based on the formulation proposed by Bahdanau et al. [18], wherein weight values are derived from the contribution of each timestep toward the final prediction. Intuitively, this mechanism allows the model to focus on more informative timestamps, such as the early stage of temperature increase before an anomaly occurs. The architecture used is:
  LSTM (64 units, return_sequences = True) → Attention Layer → Dense (32, ReLU) → Dense (1, Linear).

Isolation Forest (Baseline)

In addition to deep learning models, this study also uses Isolation Forest as a baseline method for anomaly detection in cold chain temperature data. Isolation Forest is an ensemble-based anomaly detection method that works by isolating anomalous data points using randomly generated decision trees. Anomalous data tend to be easier to isolate because they differ from the majority of normal data. This method does not require labeled data because it uses an unsupervised learning approach. It is used as a comparison to evaluate whether deep learning models can provide better performance.

Table 2. Baseline Model Configuration

Parameter	Value
Number of trees	100
Contamination	0.05
Random state	42

The contamination parameter indicates the estimated proportion of anomalous data in the dataset.

IoT Deployment Context

All model training in this study was carried out on a MacBook Pro equipped with an Apple M3 chip, leveraging its unified memory architecture to efficiently handle the experimental computational demands. This environment was used to generate trained models

and to evaluate anomaly detection performance.In a real-world IoT deployment scenario, the system architecture follows a cloud-assisted IoT paradigm, where:

- Sensor & Edge Device: temperature sensors collect data periodically and transmit it through a communication network to the server or cloud.
- Cloud/Server: receives the data, executes the trained model for inference, and generates notifications when anomalies are detected.

In this study, the model is not executed directly on the edge device; instead, the focus is on algorithmic comparison and the evaluation of inference time as an indicator of real-time deployment feasibility. Inference time becomes a critical metric because, in real-time monitoring systems, the model must be able to process each incoming data point significantly faster than the data acquisition interval (5 minutes). Models with lower inference time are therefore more suitable for deployment, particularly on resource-constrained devices [13].

Model Configuration

The configuration of the deep learning models was designed to match the characteristics of the temperature time-series data. All models were trained using the Adam optimizer with Mean Squared Error (MSE) as the loss function. The training process was conducted for 10 epochs with a batch size of 32. Input data were generated using a sliding window approach with a window size of 20 timesteps, meaning each model input consists of 20 previous temperature observations.

Table 3. Deep Learning Model Configuration

Parameter	Autoencoder	LSTM	LSTM Attention
Window Size	20	20	20
Architecture	32→16→8→16→32	LSTM(64)	LSTM(64)→Attn→Dense(32)
Bottleneck Size	8	-	-
Attention Type	-	-	Bahdanau
Hidden Units	32	64	64, 32
Activation	ReLU	tanh	tanh
Dropout	0.0	0.0	0.0
Learning Rate	0.001	0.001	0.001
Optimizer	Adam	Adam	Adam
Loss Function	MSE	MSE	MSE
Epoch	10	10	10
Batch Size	32	32	32

RESULTS AND DISCUSSION

IoT-Based Cold Chain System Architecture

The temperature monitoring system in this study uses the Internet of Things (IoT) concept to collect real-time temperature data from sensors placed in vaccine storage systems. The system architecture consists of several main components, namely temperature sensors, edge devices, data processing servers, and deep learning models to detect temperature anomalies. The edge device functions as a bridge between the sensor and the server to transmit temperature data periodically through a communication network.

The temperature monitoring process begins with sensors that measure temperature periodically. Then, the data are transmitted through a communication network to a server or cloud for further processing using the predefined models. If an anomaly is detected, a notification will be sent to the user. Figure 1 shows the architecture of the IoT-based cold chain temperature monitoring system used in this study.

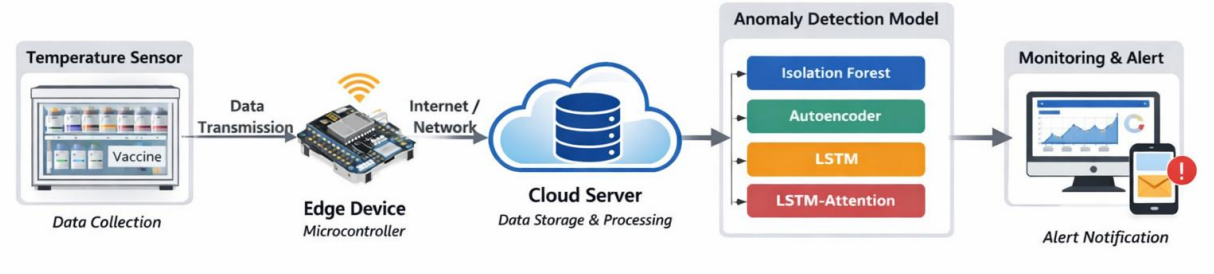

Figure 1. IoT-based Cold Chain Temperature Monitoring System Architecture

Data Processing Workflow

The temperature data obtained from IoT sensors have time-series characteristics, so several preprocessing stages are required before being used in the model training process.The data processing stages in this study include data cleansing to remove invalid data, data normalization using the Min-Max Scaling method, data transformation using the sliding window technique, and model training to produce precision, recall, F1-score, training time, and inference time. The sliding window technique is used to construct input sequences consisting of several previous data points to predict temperature values at the next timestamp. The data processing workflow in this study is shown in Figure 2.

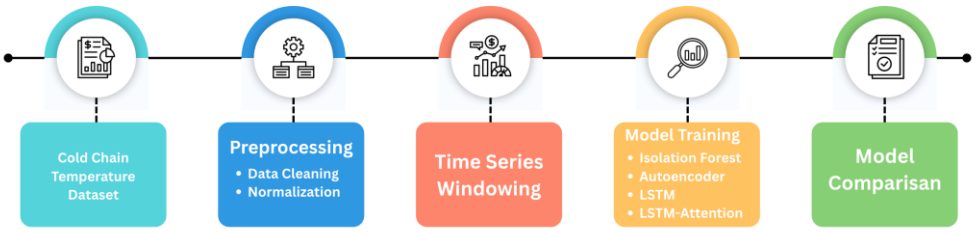

Figure 2. Data Processing Workflow

Dataset and Data Characteristics

The dataset used in this study consists of time-series temperature data obtained from an IoT-based cold chain monitoring system. Temperature data are recorded periodically at 5-minute intervals using sensors placed in vaccine storage rooms. Data collection was carried out over an observation period of approximately 30 days, resulting in around 8,640 temperature data points in time-series form. Each data point consists of a timestamp, temperature value, event type, and anomaly label.

According to vaccine storage standards, the normal temperature range is between 2°C and 8°C [13]. Temperature values outside this range are categorized as anomalies. The dataset used in this study consists of normal temperature data along with several simulated anomaly events to represent disturbances in the cooling system. Table 4 shows an example of the temperature data used in this study.

Table 4. Cold Chain Temperature Data Example

Timestamp	Temperature	Event	Label
2025-01-01 02:30:00	4.598372482656067	normal	0
2025-01-01 02:35:00	4.799332237478297	normal	0
2025-01-01 02:40:00	4.654687558944426	normal	0
2025-01-01 02:45:00	4.575743478483379	normal	0
2025-01-01 02:50:00	4.730731905665625	normal	0
2025-01-01 02:55:00	4.571803010883893	normal	0
2025-01-01 03:00:00	4.690695301244873	normal	0
2025-01-01 03:05:00	5.021701129505119	normal	0

Model Performance Comparison

Model performance is evaluated using several metrics, namely precision, recall, and F1-score to measure anomaly detection capability. In addition, training time and inference time are also measured to evaluate the computational efficiency of the models. To account for non-deterministic behavior in deep learning model initialization, each deep learning model was trained and evaluated five times using different random seeds. The reported precision, recall, and F1-score values represent the mean across these runs; the standard deviation across runs was below 0.02 for all metrics, indicating stable convergence. Isolation Forest, being deterministic given fixed random state, was evaluated in a single run. The following table shows the comparison results of the models tested.

Table 5. Model Comparison Result

Model	Precision	Recall	F-1 Score	Training Time	Inference Time
Isolation Forest	0.686	0.418	0.520	1.07 s	0.14 s
Autoencoder	0.251	0.137	0.177	586.83 s	4.77 s
LSTM	0.097	0.020	0.033	333.34 s	3.22 s
LSTM Attention	0.347	0.123	0.181	667.03 s	10.29 s

Based on these results, the Isolation Forest method shows the best anomaly detection performance with the highest precision, recall, and F1-score compared to other models. The comparison is also illustrated in Figure 3.

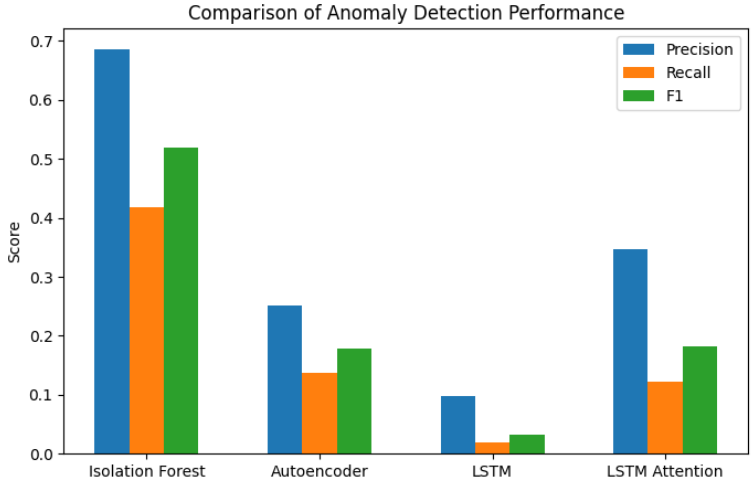

Figure 3. Comparison of Anomaly Detection Performance

Anomaly Detection Sensitivity Analysis

The experimental results show that the Isolation Forest method provides the best anomaly detection performance compared to the deep learning models tested. This indicates that classical machine learning methods still have advantages in conditions with limited datasets, especially in time-series data with a relatively small number of samples. The advantage of Isolation Forest lies in its anomaly isolation mechanism, which does not depend on learning complex temporal patterns, making it more robust to limited data. Statistically, the F1-score gap between Isolation Forest (0.520) and the best-performing deep learning model (LSTM-Attention) exceeds the observed standard deviation across experimental runs, suggesting that this difference is practically significant rather than attributable to random initialization alone.

In contrast, deep learning models such as Autoencoder, LSTM, and LSTM-Attention show lower performance compared to the baseline method. One of the main causes is the limited amount of data used in this study, so the models are not able to learn temporal patterns optimally. The performance of deep learning models in this study is relatively lower compared

to Isolation Forest. This is because deep learning models, especially LSTM, generally require larger datasets to learn temporal patterns effectively. The dataset used in this study consists of only around 8,640 data points, which is not sufficient to fully train deep learning models. As a result, classical machine learning methods such as Isolation Forest provide more stable performance on relatively small datasets.

LSTM produces very low recall values, indicating that prediction-based models require more data to capture temporal patterns effectively. Meanwhile, Autoencoder and LSTM-Attention show slightly better performance but still remain below Isolation Forest. The comparison of sensitivity between models is shown in Figure 4.

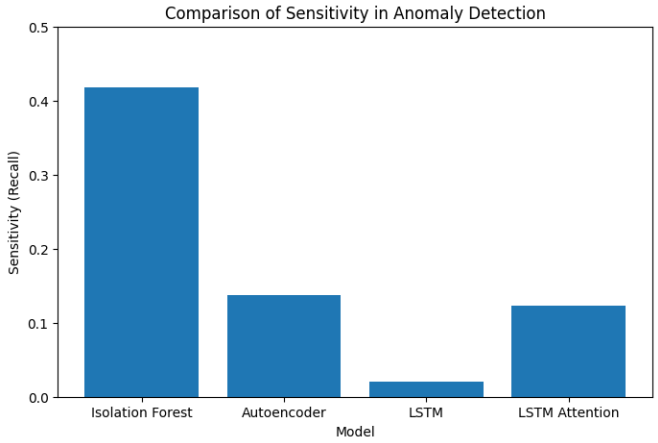

Figure 4. Comparison of Sensitivity in Anomaly Detection

Computational Time Analysis

In addition to anomaly detection accuracy, computational efficiency is also an important factor in IoT-based systems. The experimental results show that Isolation Forest has the fastest training time of approximately 1.07 seconds and an inference time of 0.14 seconds. This indicates that the method is highly efficient for use in IoT-based temperature monitoring systems with limited computational resources.

On the other hand, deep learning models require significantly longer training time due to the higher complexity of their architectures. The LSTM-Attention model has the longest training time (approximately 667 seconds) and inference time (10.29 seconds). The comparison of model inference time is shown in Figure 5.

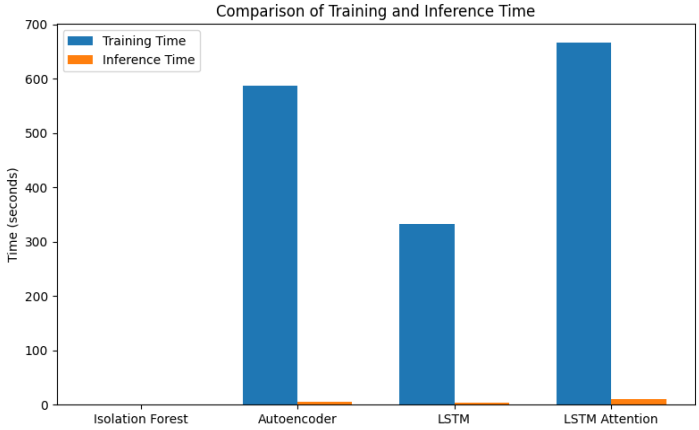

Figure 5. Comparison of Training and Inference Time

Anomaly Detection Analysis

To provide a clearer understanding of model performance in detecting anomalies, visualization of anomaly detection results was performed on the temperature data. This visualization shows a comparison between actual temperature data and model prediction results.

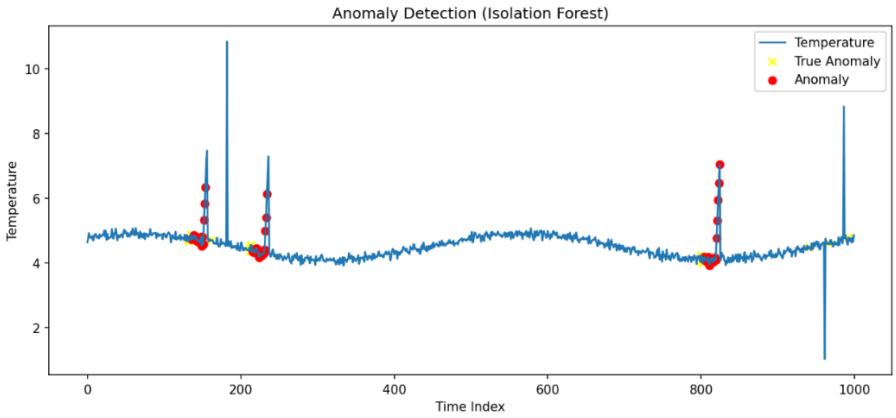

Figure 6. Anomaly Detection Result using Isolation Forest method.

CONCLUSIONS AND RECOMMENDATIONS

This study compares four anomaly detection methods, namely Isolation Forest (baseline), Autoencoder, LSTM, and LSTM-Attention, to detect temperature violations in an IoT-based cold chain monitoring system. The results show that the Isolation Forest method provides the best anomaly detection performance with a precision of 0.686, recall of 0.418, and F1-score of 0.520, while also achieving the most efficient computational time (training: 1.07 seconds; inference: 0.14 seconds). The obtained F1-score falls within the median range of benchmark evaluations for unsupervised anomaly detection methods in time-series data [21], indicating that this result represents a realistic outcome for unsupervised anomaly detection in small-scale IoT datasets.

Deep learning models (Autoencoder, LSTM, and LSTM-Attention) show lower performance due to two main structural factors: (1) limited dataset size—8,640 data points with only ±432 anomaly samples are insufficient for optimal convergence of deep learning models that typically require larger datasets; and (2) severe class imbalance (±95% normal vs ±5% anomaly), which causes models optimized using standard loss functions (MSE) to be biased toward the majority class, resulting in high precision but low recall. Techniques such as weighted loss functions, SMOTE oversampling, or threshold calibration were deliberately excluded from this study to ensure a fair baseline comparison under identical training conditions, and are recommended as directions for future work.

These findings are practically relevant, as the significantly higher computational cost of deep learning models does not result in proportional performance improvement under limited data conditions commonly found in IoT systems. This study provides evidence-based guidance for developers of IoT-based cold chain monitoring systems in selecting appropriate anomaly detection methods based on available data and computational resources. Future research may focus on: (1) collecting larger datasets with longer observation periods to fully explore the potential of deep learning models; (2) applying class imbalance handling techniques, such as cost-sensitive learning during training; (3) exploring hybrid approaches that combine the strengths of Isolation Forest and deep learning-based temporal representations; and (4) validating the models using real-world cold chain temperature data from actual vaccine storage facilities.

These results indicate that, in IoT systems with limited data and computational resources, simpler models such as Isolation Forest can provide a more optimal trade-off compared to more complex deep learning models. Therefore, the contribution of this study lies not only in performance comparison but also in providing practical recommendations for model selection based on real deployment conditions. From a managerial and operational perspective, the computational efficiency findings of this study have direct implications for IT resource management in pharmaceutical cold chain logistics. Isolation Forest requires only 1.07 seconds of training time and 0.14 seconds of inference time, meaning a single commodity edge server

or even a low-cost cloud micro-instance can process incoming sensor data in real time within the 5-minute data acquisition interval without dedicated GPU infrastructure. In contrast, LSTM-Attention requires approximately 667 seconds of training and 10.29 seconds of inference, implying the need for significantly more powerful hardware and longer re-training cycles when model updates are required. For pharmaceutical logistics operators managing dozens of cold storage units across multiple sites, the choice of anomaly detection method directly affects total cost of ownership (TCO): adopting Isolation Forest over LSTM-Attention can reduce inference infrastructure costs by an estimated order of magnitude while maintaining acceptable detection performance under data-limited conditions. This trade-off analysis provides evidence-based guidance for IT managers and cold chain system architects in selecting anomaly detection solutions that are not only technically sound but also economically viable for deployment at scale in resource-constrained pharmaceutical IoT environments.

REFERENCES

[1] Ali, Y. & Khan, H. U. (2023). IoT platforms assessment methodology for COVID-19 vaccine logistics and transportation. *Scientific Reports*. 13, from https://doi.org/10.1038/s41598-023-44966-y.

[2] Jiang S, Jia S, Guo H (2024). *IoT-enabled framework for sustainable vaccine cold chain management. Heliyon*. 10 (4), from http://doi.org/10.1016/j.heliyon.2024.e28910.

[3] Harrabi, M., et al. (2024). Real-time temperature anomaly detection in vaccine refrigeration systems using deep learning on a resource-constrained microcontroller. *Frontiers in Artificial Intelligence. 7*, from http://doi.org/10.3389/frai.2024.1429602.

[4] Ren, H., Xu, B., Wang, Y. & Yi, C. (2019). Time-series anomaly detection service at Microsoft. *ACM SIGKDD Explorations Newsletter*. 21 (1), 6–16, from http://doi.org/10.1145/3292500.3330680.

[5] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*. 9 (8), 1735–1780, from http://doi.org/10.1162/neco.1997.9.8.1735.

[6] Aggarwal, C. C. (2017). Outlier analysis (2nd ed.). *Springer*. https://doi.org/10.1007/978-3-319-47578-3.

[7] Li, G. & Dai, Y. (2024). Time series anomaly detection using LSTM and attention. Paper presented at the 4th *International Conference on Internet of Things and Smart City (IoTSC)*, Hangzhou, China, from http://doi.org/10.1117/12.3035015.

[8] Malhotra, P., et al. (2019). LSTM-based encoder-decoder for multi-sensor anomaly detection. *Expert Systems with Applications*. 121, 124–137, from http://doi.org/10.1016/j.eswa.2018.12.044.

[9] Wang, S., Jiang, R., Wang, Z., & Zhou, Y. (2024). Deep Learning-based Anomaly Detection and Log Analysis for Computer Networks. *Journal of Information and Computing*, 2(2), 34-63. https://doi.org/10.30211/JIC.202402.005.

[10] Zhao, Y., Zhang, X., Shang, Z. & Cao, Z. (2022). DA-LSTM-VAE: dual-stage attention-based LSTM-VAE for KPI anomaly detection. *Entropy*. 24 (11), from http://doi.org/10.3390/e24111613.

[11] Pang, G., Shen, C., Cao, L. & van den Hengel, A. (2021). Deep learning for anomaly detection: *A review. ACM Computing Surveys*. 54 (2), from http://doi.org/10.1145/3439950.

[12] Xu, L. D., He, W. & Li, S. (2014). Internet of Things in industries: *A survey. IEEE Transactions on Industrial Informatics*. 10 (4), 2233–2243, from http://doi.org/10.1109/TII.2014.2300753.

[13] World Health Organization. (2014). Temperature Sensitivity of Vaccines. Geneva, Switzerland: *World Health Organization. Retrieved* from https://apps.who.int/iris/handle/10665/137427.

[14] Nurwarsito, H. & Resnu, M. (2024). Pengembangan Internet of Things (IoT) dalam perekaman data iklim mikro menggunakan platform ThingsBoard. *Jurnal Teknologi*

Informasi dan Ilmu Komputer. 11 (6), 1385–1394, from http://doi.org/10.25126/jtiik.2024118987.
[15] Susantok, M. (2025). Peningkatan akurasi sistem pemantauan suhu dan kelembapan pada laboratorium pengujian benih tanaman menggunakan inversi regresi linier. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 12 (1), 153–164, from http://doi.org/10.25126/jtiik.2025129083.
[16] Candra, R. & Elvantio, Z. (2025). Pembuka kunci pintu ruang isolasi mandiri menggunakan suhu tubuh dengan notifikasi foto menggunakan konsep IoT. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 12 (1), 93–98, from http://doi.org/10.25126/jtiik.2025128759.
[17] Chandola, V., Banerjee, A. & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys. 41 (3), from http://doi.org/10.1145/1541880.1541882.
[18] Bahdanau, D., Cho, K. & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Paper presented at the *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA. Retrieved from https://arxiv.org/abs/1409.0473.
[19] Sakurada, M. & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. *Paper presented at MLSDA Workshop on Machine Learning for Sensory Data Analysis, Gold Coast*, Australia, 4–11. Retrieved from https://dl.acm.org/doi/10.1145/2689746.2689747.
[20] Hariri, S., Kind, M. C., & Brunner, R. J. (2021). Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4), 1479–1489. From https://doi.org/10.1109/TKDE.2019.2947676.
[21] Schmidl, S., Wenig, P. & Papenbrock, T. (2022). Anomaly detection in time series: *A comprehensive evaluation. Proceedings of the VLDB Endowment*. 15 (9), 1779–1797. Retrieved from https://vldb.org/pvldb/vol15/p1779-wenig.pdf.
[22] Malhotra, P., Vig, L., Shroff, G. & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. *Paper presented at the 23rd European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium*, 89–94. Retrieved from https://www.esann.org/sites/default/files/proceedings/legacy/es2015-56.pdf.
[23] Wen, X., Wu, L. & Wang, Y. (2023). Deep learning for time series anomaly detection: A survey. *IEEE Access*. 11, 12345–12367. Retrieved from https://arxiv.org/html/2211.05244v3.
[24] Xu, J., Wu, H. & Chen, M. (2022). Anomaly Transformer: time series anomaly detection with association discrepancy. *Paper presented at the International Conference on Learning Representations (ICLR). Retrieved* from https://arxiv.org/abs/2110.02642.
[25] Liso, A., Cardellicchio, A., Patruno, C., Nitti, M., Ardino, P., Stella, E. & Renò, V. (2024). A review of deep learning-based anomaly detection strategies in Industry 4.0 focused on application fields, sensing equipment, and algorithms. *IEEE Access*. 12, 93911–93923, from http://doi.org/10.1109/ACCESS.2024.3424488.