SJER Scientific Journal of Engineering Research

futuristech PT. Teknologi Futuristik Indonesia

Received: February 2, 2026; Revised: May 17, 2026; Accepted: May 20, 2026; Published: May 22, 2026; Current version: May 22, 2026. e-ISSN: 3109-1725
Digital Object Identifier 10.64539/sjer.v2i3.2026.432

Article

Deep Learning–Driven Anomaly Detection for IoT-Enabled Smart Engineering Systems

Godfrey Perfectson Oise[1,*], Kevin Chinedu Pius[1], Felix Oshiorenoya Uloko[2], Immunhierokene Clinton Obrorindo[3], Roli Lydia Oshasha[3]

[1] Department of Computing, Wellspring University, Benin City, Edo State. Nigeria; e-mail: godfrey.oise@wellspringuniversity.edu.ng (G. P. Oise).
[2] Department Computer Science, Veritas University, Abuja, Nigeria.
[3] Department of Computer Science and Information Technology, Petroleum Training Institute, Effurun Delta State, Nigeria.
* Correspondence Author

The authors received no financial support for the research, authorship, and/or publication of this article.

Abstract: The rapid adoption of Internet of Things (IoT) technologies in smart engineering systems has increased the need for reliable anomaly detection mechanisms capable of identifying cyberattacks, operational faults, and abnormal system behaviors in complex cyber–physical environments. Existing rule-based and conventional machine learning approaches often struggle to effectively model the non-linear, high-dimensional, and highly imbalanced nature of IoT-generated multivariate time-series data, thereby limiting their capability to detect subtle and previously unseen anomalies. To address these challenges, this study proposes a deep learning–driven anomaly detection framework based on a hybrid CNN–LSTM autoencoder architecture for modeling spatiotemporal system behavior in IoT-enabled engineering environments. The proposed framework integrates convolutional neural networks for spatial feature extraction with long short-term memory networks for temporal dependency learning, while anomaly detection is performed using reconstruction error analysis and adaptive thresholding under unsupervised learning conditions. Experimental evaluation was conducted using the BATADAL-A dataset, which represents a realistic cyber–physical water distribution system. The results demonstrate stable convergence and strong generalization performance, with closely aligned training and validation losses throughout the learning process. The proposed framework achieved 90% overall accuracy, anomaly precision of 0.83, anomaly recall of 0.22, and an AUC of 0.677, indicating effective modeling of normal operational behavior but limited sensitivity to rare anomalous events. These findings demonstrate that the proposed CNN–LSTM autoencoder provides reliable low–false alarm monitoring for IoT-enabled smart engineering systems while highlighting the need for future improvements to enhance anomaly sensitivity and robustness in safety-critical applications.

Keywords: Anomaly detection; IoT-enabled engineering systems; CNN–LSTM autoencoder; Cyber–physical systems; Time-series analysis.

1. Introduction

The rapid proliferation of Internet of Things (IoT) technologies has fundamentally reshaped modern engineering systems by enabling pervasive sensing, real-time data acquisition, and intelligent automation across critical infrastructures such as smart grids, water distribution networks, industrial control systems, intelligent transportation platforms, and smart manufacturing environments [1]. These systems increasingly rely on dense networks of heterogeneous, resource-constrained devices that continuously generate high-dimensional, multivariate time-series data reflecting complex cyber–physical interactions. While such connectivity enhances operational efficiency and situational awareness, it also significantly expands the system attack surface and introduces new failure modes arising from cyber intrusions, equipment degradation, control malfunctions, and unexpected environmental disturbances [2]. The tightly coupled nature of cyber and physical

components in IoT-enabled engineering systems implies that even subtle anomalies can propagate rapidly, leading to cascading failures, safety hazards, and substantial economic losses. Consequently, reliable and adaptive anomaly detection has become a critical requirement for ensuring the security, stability, and resilience of smart engineering infrastructures [3].

IoT systems are widely used but highly vulnerable to cyberattacks, and their heterogeneous and noisy data characteristics make anomaly detection particularly challenging. To address these challenges, this study proposes a deep learning–based anomaly detection framework using a hybrid CNN–LSTM autoencoder to learn robust spatiotemporal representations for identifying anomalous IoT behaviors [4]. Conventional anomaly detection and intrusion detection mechanisms in engineering systems have traditionally relied on rule-based logic, statistical thresholds, and signature-driven models derived from expert knowledge of system dynamics.

Although effective in controlled or stationary environments, these approaches struggle to cope with the nonlinearity, non-stationarity, and scale of contemporary IoT-generated data [5], [6]. Their dependence on predefined patterns limits their ability to detect previously unseen attacks or evolving fault conditions, while manual rule maintenance becomes impractical as system complexity grows. To overcome these limitations, data-driven machine learning approaches have been increasingly adopted, offering improved adaptability and the ability to learn system behavior directly from operational data. However, classical machine learning techniques typically depend on handcrafted features and shallow representations, which may fail to capture the intricate spatiotemporal dependencies inherent in large-scale cyber–physical systems.

Recent advances in deep learning have provided powerful tools for modeling complex, high-dimensional data streams in IoT-enabled engineering environments [7]. Deep architectures such as autoencoders, convolutional neural networks, recurrent neural networks, and long short-term memory networks have been extensively investigated for fault diagnosis and anomaly detection in industrial and cyber–physical systems [8]. Autoencoder-based methods, in particular, have gained prominence due to their suitability for unsupervised learning scenarios, where labeled attack or fault data are scarce or incomplete [9], [10]. By learning compact latent representations of normal system behavior and identifying deviations through reconstruction error, these models offer inherent robustness against zero-day attacks and unknown anomalies.

Convolutional neural networks have been shown to effectively capture spatial correlations and local dependencies among sensor variables [11], [12], while LSTM-based models excel at learning long-range temporal dependencies and system dynamics in sequential data. Consequently, hybrid CNN–LSTM architectures have emerged as a promising paradigm for jointly modeling spatial and temporal characteristics of complex engineering processes [13].

Despite the growing body of literature, several fundamental challenges remain unresolved. First, many existing deep learning–based anomaly detection studies emphasize classification accuracy without sufficiently considering the operational constraints of real-world engineering systems, such as computational complexity, latency, and deployment feasibility on edge or embedded platforms. Second, severe class imbalance in engineering datasets often results in models that achieve high overall accuracy while exhibiting poor sensitivity to rare but safety-critical anomalies, thereby undermining their practical utility [14]. Third, a significant portion of prior work treats anomaly detection as an isolated data analytics task, with limited discussion of how detection outputs integrate into broader system monitoring, control, and decision-support workflows. Furthermore, the interpretability and stability of deep models remain open concerns, particularly in safety-critical domains where engineers must understand and trust automated decisions [15], [16].

In response to these limitations, this study proposes a deep learning–driven anomaly detection framework specifically designed for IoT-enabled smart engineering systems, with a focus on spatiotemporal modeling and practical deployability. The proposed approach employs a hybrid CNN–LSTM autoencoder architecture trained exclusively on normal operational data to learn a high-fidelity representation of baseline system dynamics.

Multivariate sensor and actuator data are modeled as structured temporal sequences, enabling the convolutional layers to extract inter-sensor relationships and the LSTM layers to encode long-term temporal dependencies [17], [18]. Anomalies arising from cyberattacks, system faults, or abnormal operational conditions are detected using reconstruction error–based analysis, enabling adaptive detection without relying on predefined signatures or labeled attack samples [19], [20]. The methodology is evaluated using a realistic cyber–physical system dataset that reflects the complexity and non-stationarity of real engineering environments, and performance is assessed using both detection metrics and system-level analysis.

This study contributes to the field of IoT-enabled smart engineering systems through the following key aspects:

- Development of a hybrid CNN–LSTM autoencoder for spatiotemporal representation learning in cyber–physical anomaly detection tasks.
- Design of an unsupervised reconstruction-error-based detection framework suitable for environments with limited labeled anomaly data.
- Comprehensive evaluation using the BATADAL-A dataset under highly imbalanced conditions,

highlighting realistic performance limitations.
- Empirical analysis of the trade-off between false alarm reduction and anomaly detection sensitivity, showing the conservative nature of reconstruction-based methods.
- Identification of limitations in anomaly recall and separability, providing insights for future improvement strategies in IoT-based anomaly detection systems.

The remainder of the paper is organized as follows: Section 2 presents the materials and methods, including the dataset description and proposed model architecture; Section 3 discusses the experimental results and performance evaluation; Section 4 outlines future research directions; and Section 5 concludes the study.

2. Methodology

2.1. System Architecture and CNN–LSTM Autoencoder-Based Anomaly Detection Framework

This study proposes a deep learning–driven anomaly detection framework for IoT-enabled smart engineering systems by integrating data science, artificial intelligence, and cybersecurity principles to identify anomalous behaviors in complex cyber-physical environments. The framework is designed to be generic yet adaptable across domains such as energy systems, industrial automation, water distribution networks, and intelligent infrastructure, supporting both simulation-based analysis and practical deployment. At its core, the approach employs a CNN–LSTM autoencoder that captures spatial inter-sensor relationships through convolutional layers and long-term temporal dependencies via LSTM layers, enabling robust spatiotemporal representation learning. As illustrated in Figure 1, the model compresses normal operational behavior into a latent space and reconstructs input sequences, where anomalies are detected based on reconstruction error; higher deviations indicate faults, cyberattacks, or control irregularities. While this hybrid architecture improves detection performance compared to simpler models, it also introduces higher computational complexity and requires careful hyperparameter tuning.

The system follows a layered IoT architecture in which distributed sensors and actuators generate multivariate time-series data representing system states, control signals, and operational metrics. These data are transmitted to edge or centralized computing nodes, where preprocessing—such as normalization, missing value handling, noise reduction, and temporal segmentation—is performed prior to analysis. Feature extraction and anomaly detection are then conducted using the CNN–LSTM autoencoder, and detected anomalies are forwarded to a system-level evaluation module to assess their impact on reliability, stability, and operational efficiency. This modular design ensures scalability and interpretability in real-world engineering environments.

For implementation, the model is trained exclusively on normal operational data using a sliding window approach that converts continuous sensor streams into fixed-length sequences. The encoder applies one-dimensional convolutional layers to capture local spatial patterns and inter-sensor correlations, followed by stacked LSTM layers to model temporal dependencies, while the decoder mirrors this structure to reconstruct inputs. Training is performed via backpropagation through time using an adaptive optimizer, with mini-batch optimization, early stopping, and regularization to ensure stable convergence and prevent overfitting. A validation subset of normal data is used for performance monitoring and hyperparameter tuning. The framework leverages GPU acceleration during training for efficiency, while inference is evaluated under resource-constrained conditions to reflect realistic deployment scenarios. It is also designed with edge-assisted deployment considerations to align with practical constraints such as latency, bandwidth, and energy efficiency, although detailed latency and throughput benchmarking is reserved for future work. The overall workflow of the proposed anomaly detection framework is summarized in Algorithm 1.

2.2. Anomaly Scoring and Decision Thresholding

Anomaly detection is achieved by computing the reconstruction error between each input sequence and its reconstructed output. A dynamic decision threshold is derived from the statistical distribution of reconstruction errors observed during normal operation. Sequences whose reconstruction errors exceed this threshold are classified as anomalous. This adaptive thresholding mechanism enables sensitivity to evolving system behavior and avoids reliance on fixed thresholds that may be unsuitable for non-stationary engineering environments. Detected anomalies are time-stamped and correlated with system states to support root-cause analysis and informed engineering decision-making.

The anomaly threshold parameter θ plays a critical role in balancing detection sensitivity and false alarm rate. Lower threshold values increase anomaly sensitivity but may produce excessive false positives, whereas higher thresholds reduce false alarms at the expense of missed anomalies. In this study, θ was derived from the reconstruction error distribution of normal operational data to maintain stable and conservative detection behavior.

2.3. Dataset Description

The experimental evaluation of the proposed framework is conducted using the BATADAL-A dataset, a widely recognized benchmark for cyberattack and anomaly detection in cyber-physical and engineering systems [21]. The dataset consists of multivariate time-series data collected from a realistic water distribution testbed that closely resembles real-world industrial control and smart

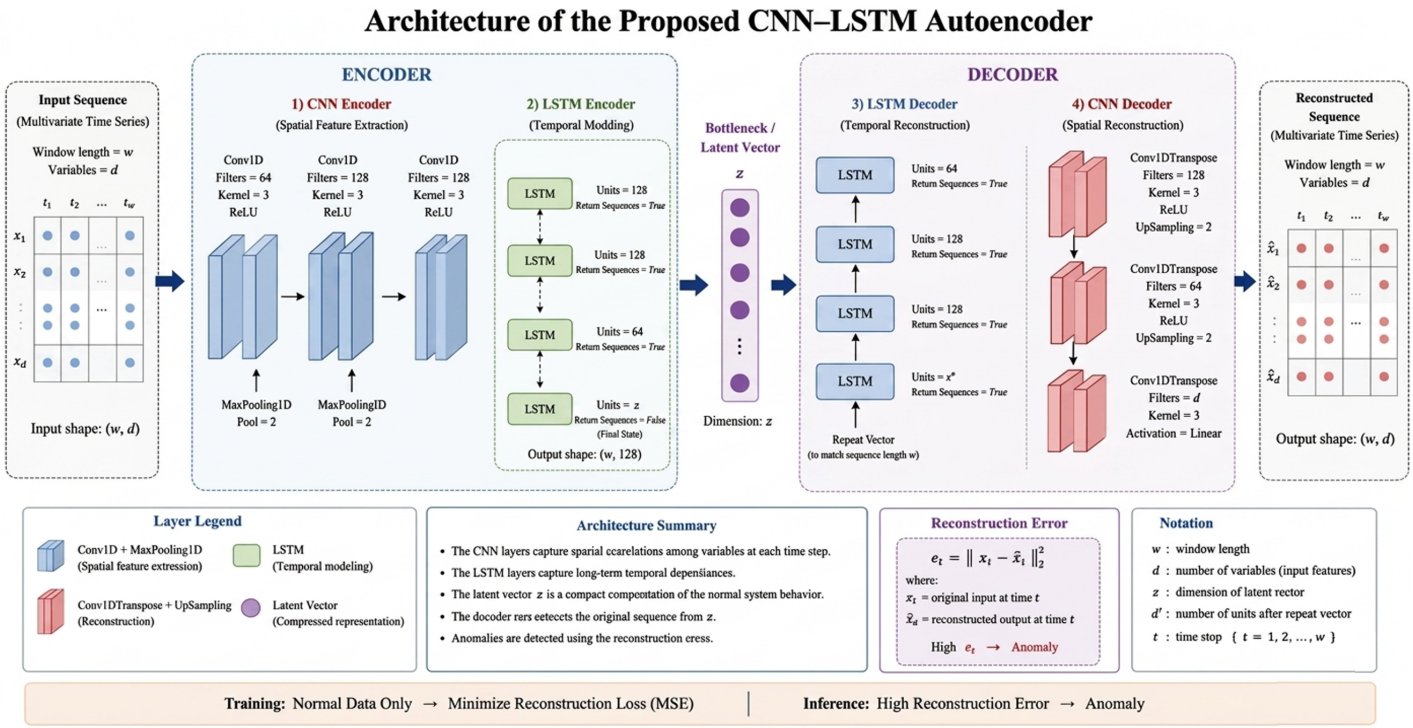

Figure 1. Proposed CNN–LSTM Autoencoder Architecture.

Algorithm 1. CNN–LSTM Autoencoder–Based Anomaly Detection.

Multivariate IoT time-series data $X \in \mathbb{R}^{T \times n}$, *window length* w, *trained CNN–LSTM Autoencoder model, anomaly threshold* θ *Anomaly labels* $A \in \{0,1\}$ *for each time window.*

Step 1: *Data Preprocessing. Normalize sensor and actuator data to a fixed range. Handle missing values using interpolation or forward filling. Segment the continuous data into overlapping windows:*

$$X^{(i)} = \{x_i, x_{i+1}, \dots, x_{i+w-1}\} \quad (1)$$

Step 2: *Model Training (Normal Data Only) Initialize CNN–LSTM Autoencoder parameters. Extract spatial features using convolutional layers. Capture temporal dependencies using LSTM layers. Reconstruct the input window using the decoder. Minimize reconstruction loss via backpropagation. Repeat until convergence or early stopping criteria are met.*

Step 3: *Anomaly Detection (Inference Phase) Compute reconstructed output* $\hat{X}^{(j)}$ *Calculate reconstruction error* $E^{(j)}$ *Label window as anomalous* ($A^{(j)} = 1$) *Label window as normal* ($A^{(j)} = 0$).

Step 4: *Post-Processing Aggregate anomaly decisions across time. Correlate detected anomalies with engineering system states.*

infrastructure environments. It includes timestamped measurements of physical variables such as tank levels, flow rates, pump statuses, valve states, and pressure readings. Each data instance is labeled with an anomaly indicator that distinguishes normal system operation from abnormal behavior resulting from cyberattacks or system disruptions. Although labeled data are available, only normal operational data are used for model training to reflect a realistic unsupervised learning setting, while labeled data are used exclusively for evaluation purposes.

The dataset contains multivariate sensor and actuator measurements collected under both normal and attack scenarios. Approximately 70% of normal operational data are used for training and validation, while the remaining data, consisting of both normal and anomalous samples, are reserved for testing. The sequence window length is experimentally selected to balance temporal dependency learning and computational efficiency.

2.4. Data Preprocessing

Before training, the dataset undergoes several preprocessing steps to ensure data quality and learning stability. All numerical features are normalized using min–max scaling to eliminate bias caused by differing measurement scales. Missing values, where present, are handled using interpolation methods to preserve temporal continuity. The continuous time-series data are segmented into overlapping windows of fixed length using a sliding window strategy. This approach enables the model to capture temporal dependencies and evolving system behavior. Feature redundancy and noise are further reduced using correlation analysis and variance-based filtering, ensuring that the deep learning model focuses on informative system dynamics while maintaining computational efficiency.

2.5. Mathematical Model

2.5.1. Input Representation

Let:

$$X = \{x_1, x_2, \dots, x_T\}, \quad x_t \in \mathbb{R}^n \tag{2}$$

represent multivariate sensor readings collected from an IoT-enabled smart engineering system, where T denotes the total number of time steps and n is the number of sensor and actuator variables. The continuous time-series data are segmented into overlapping windows of fixed length w such that:

$$X^{(i)} \in \mathbb{R}^{w \times n} \tag{3}$$

2.5.2. Encoder Function

The encoder learns a compact latent representation for each input window:

$$z^{(i)} = f_{\text{enc}}\left(X^{(i)}; \theta_{\text{enc}}\right) \tag{4}$$

where $f_{\text{enc}}(\cdot)$ denotes the encoder function parameterized by θ_{enc}. The encoder consists of convolutional layers for spatial feature extraction across sensor dimensions, followed by long short-term memory (LSTM) layers for modeling temporal dependencies in the data.

2.5.3. Decoder Function

The decoder reconstructs the original input window from the latent representation:

$$\hat{X}^{(i)} = f_{\text{dec}}\left(z^{(i)}; \theta_{\text{dec}}\right) \tag{5}$$

where $f_{\text{dec}}(\cdot)$ mirrors the encoder architecture and is parameterized by θ_{dec}. The decoder aims to accurately reconstruct normal system behavior captured in the latent space.

2.5.4. Autoencoder Reconstruction Loss

The CNN–LSTM Autoencoder is trained by minimizing the reconstruction loss over normal operational data:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \left\| X^{(i)} - \hat{X}^{(i)} \right\|_2^2 \tag{6}$$

where N denotes the number of training windows and $\|\cdot\|_2$ represents the Euclidean norm. Minimizing this loss enforces accurate reconstruction of normal system dynamics while amplifying reconstruction errors for anomalous patterns.

2.5.5. Anomaly Score Definition

For each input window, an anomaly score is computed based on the reconstruction error:

$$S^{(i)} = \frac{1}{w \cdot n}\sum_{t=1}^{w}\sum_{k=1}^{n}\left(x_{t,k}^{(i)} - \hat{x}_{t,k}^{(i)}\right)^2 \tag{7}$$

Higher values of $S^{(i)}$ indicate greater deviation from the learned normal behavior of the engineering system.

2.5.6. Anomaly Decision Rule

A window is classified as anomalous according to the following decision rule:

$$A^{(i)} = \begin{cases} 1, & \text{if } S^{(i)} > \theta \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where the anomaly threshold θ is determined statistically from the reconstruction error distribution of the training data:

$$\theta = \mu_S + \lambda\sigma_S \tag{9}$$

here, μ_S and σ_S denote the mean and standard deviation of reconstruction errors under normal operation, respectively, and λ is a sensitivity parameter that controls the trade-off between detection sensitivity and false alarms.

2.6. Evaluation Metrics

The proposed framework is evaluated using standard anomaly detection metrics, including precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve (AUC-ROC). In addition to detection performance, computational metrics such as inference latency and memory usage are also measured to assess practical feasibility. Together, these metrics provide a comprehensive evaluation of both detection effectiveness and engineering applicability.

Accuracy is defined in Equation 10, while precision and recall are defined in Equation 11 and Equation 12, respectively. The F1-score, which represents the harmonic mean of precision and recall, is given in Equation 13.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2x\frac{(Precision\ x\ Recall)}{Precision + Recall} \tag{13}$$

3. Results and Discussion

3.1. Results

Table 1 presents a CNN–LSTM hybrid for time-series modeling, designed for tasks such as cyber–physical system anomaly detection. It uses 1D convolution and max

Table 1. CNN–LSTM Autoencoder Model Architecture.

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 24, 32)	4,160
max_pooling1d_1 (MaxPooling1D)	(None, 12, 32)	0
lstm_4 (LSTM)	(None, 12, 64)	24,832
lstm_5 (LSTM)	(None, 32)	12,416
repeat_vector_1 (RepeatVector)	(None, 24, 32)	0
lstm_6 (LSTM)	(None, 24, 32)	8,320
lstm_7 (LSTM)	(None, 24, 64)	24,832
conv1d_3 (Conv1D)	(None, 24, 43)	8,299

Total params: 82,859 (323.67 KB)
Trainable params: 82,859 (323.67 KB)
Non-trainable params: 0 (0.00 B)

Table 2. Classification Report.

Class	Precision	Recall	F1-Score	Support
0	0.90	0.99	0.95	3661
1	0.83	0.22	0.35	492
Accuracy			0.90	4153
Macro Avg	0.87	0.61	0.65	4153
Weighted Avg	0.90	0.90	0.88	4153

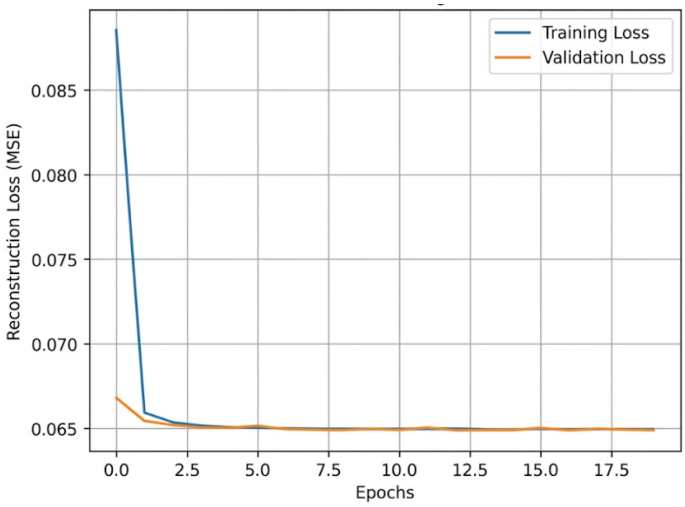

Figure 2. Training and Validation Loss.

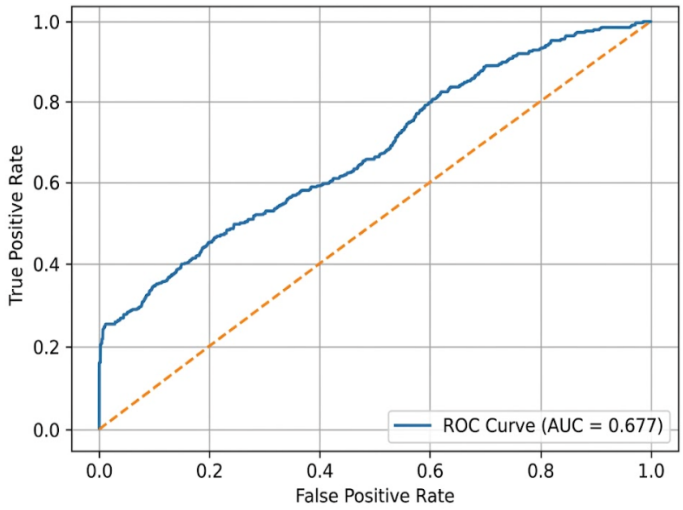

Figure 4. ROC Curve for CNN-LSTM Autoencoder.

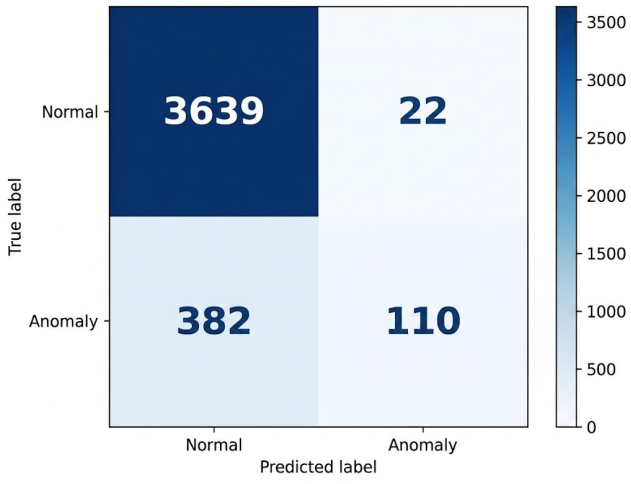

Figure 3. Confusion Matrix for Anomaly Detection.

pooling to extract local temporal features, followed by stacked LSTMs to capture short- and long-term dependencies. A RepeatVector enables sequence-to-sequence learning, and a final 1D convolution maps the features to the output. The model has 82,859 trainable parameters, primarily from the LSTM layers, making it suitable for complex multivariate temporal data.

Table 2 displays that the model performs well on the majority class (0) but struggles with the minority class (1). For class 0, precision is 0.90, recall is 0.99, and F1-score is 0.95, indicating almost all actual 0s are correctly identified. For class 1, precision is 0.83, but recall drops to 0.22, resulting in a low F1-score of 0.35, meaning most class 1 instances are missed. Overall accuracy is 90%, but the macro-average F1-score of 0.65 highlights poor performance on

the minority class. The weighted averages appear high due to class imbalance, which masks the model's difficulty in detecting class 1. Improving recall for class 1 is necessary to achieve more balanced predictions.

Figure 2 demonstrates rapid convergence, with both training and validation reconstruction losses decreasing sharply within the first few epochs and stabilizing thereafter. The close alignment between training and validation loss curves indicates strong generalization capability and absence of overfitting. This behavior confirms the model's ability to efficiently learn compact latent representations of the input sequences while maintaining stable reconstruction performance across unseen data.

Figure 3 shows that the CNN–LSTM autoencoder effectively models normal behavior, achieving a high true negative rate and low false alarm rate. However, its anomaly detection is limited, with many anomalies misclassified as normal, reflecting a conservative approach that favors specificity over sensitivity. Improving anomaly recall may require adaptive thresholds, hybrid detection methods, or enhanced latent-space regularization.

Figure 4 shows that the CNN–LSTM autoencoder achieves an AUC of 0.677, indicating moderate ability to distinguish anomalies from normal behavior. The ROC curve outperforms random classification, but overlapping reconstruction errors limit separability, suggesting that stronger discriminative mechanisms are needed to improve anomaly detection sensitivity. The moderate AUC value of 0.677 further indicates that reconstruction error distributions of normal and anomalous samples partially overlap. This overlap is common in cyber–physical systems where stealthy attacks and gradual faults may closely resemble legitimate operational behavior. Consequently, certain anomalies remain embedded within the learned representation of normal system dynamics, thereby limiting statistical separability.

3.2. Discussion

The proposed CNN–LSTM autoencoder–based anomaly detection framework was evaluated using multivariate time-series data obtained from a realistic IoT-enabled cyber–physical engineering system. The observed convergence behavior is consistent with findings reported by [3]. The close correspondence between training and validation losses indicates that the model successfully learns the underlying dynamics of normal system operation without overfitting [22], [23]. This behavior confirms the suitability of the proposed architecture for modeling complex, non-linear, and temporally dependent processes commonly encountered in smart engineering environments. Although the training and validation losses indicate stable reconstruction learning, low reconstruction loss does not necessarily imply strong anomaly separability. Certain anomalous behaviors share statistical similarities with normal operational patterns, resulting in overlapping reconstruction-error distributions and reduced ROC performance.

The classification results reveal that the framework performs strongly in identifying normal operational states, achieving high precision, recall, and F1-score for the majority class. This indicates that the CNN–LSTM autoencoder effectively captures dominant spatiotemporal patterns associated with normal system behavior and maintains a low false alarm rate.

The BATADAL-A dataset exhibits significant class imbalance, where normal operational samples substantially outnumber anomalous events. Consequently, overall accuracy alone may provide an overly optimistic assessment of detection performance because the model can achieve high accuracy by predominantly predicting the majority class. For this reason, additional metrics including recall, F1-score, and AUC were emphasized in the evaluation. The relatively low anomaly recall (0.22) indicates that many anomalous events remain undetected, reflecting the conservative behavior of reconstruction-based anomaly detection methods trained solely on normal data. Although anomaly precision remains high (0.83), the imbalance between precision and recall suggests that the model prioritizes minimizing false alarms over maximizing anomaly sensitivity. This trade-off is particularly important in safety-critical engineering systems, where missed anomalies may have severe consequences.

From an engineering perspective, this characteristic is highly desirable, as excessive false positives can lead to alarm fatigue, unnecessary maintenance actions, and reduced operator trust in monitoring systems [24]. The confusion matrix further confirms a high true negative rate, demonstrating that the model reliably distinguishes normal operational regimes from deviations. In contrast, the detection performance for anomalous events exhibits lower recall, indicating that a substantial portion of anomalies are misclassified as normal. Although the precision for detected anomalies remains relatively high, the reduced sensitivity highlights a known limitation of reconstruction-based anomaly detection methods trained solely on normal data [25]. In such approaches, anomalies that exhibit partial similarity to normal operating conditions, such as stealthy cyberattacks, gradual faults, or controlled disturbances, may not generate sufficiently large reconstruction errors to exceed the decision threshold. This effect is amplified by the inherent class imbalance present in engineering datasets, where abnormal events are rare but often critical. These findings suggest that while the proposed framework is conservative and reliable in avoiding false alarms, it may under-detect subtle or evolving anomalous behaviors [26]. The receiver operating characteristic analysis further supports these observations.

The obtained area under the curve indicates moderate discrimination capability between normal and anomalous samples. Although the reconstruction error provides

meaningful separation, overlapping error distributions suggest that certain anomalous patterns remain embedded within the learned normal behavior manifold. In cyber–physical systems, this overlap reflects the physical constraints and control mechanisms that can mask malicious or faulty behavior within acceptable operational limits. Consequently, anomaly detection in such systems remains a fundamentally challenging task, particularly when relying on unsupervised learning paradigms. From a system-level standpoint, the observed trade-off between specificity and sensitivity reflects a practical design consideration for smart engineering applications. In many critical infrastructures, maintaining operational continuity and avoiding unnecessary interventions is often prioritized over aggressive anomaly detection.

4. Future Work

The proposed framework demonstrates strong stability and low false alarm rates under reconstruction-based anomaly detection. Nevertheless, improving anomaly sensitivity, particularly for rare and subtle events, remains an important direction for future research. Potential enhancements include adaptive thresholding strategies that can adjust to evolving system dynamics, as well as the integration of prediction-based components or latent-space regularization to improve feature separability. In addition, hybrid detection schemes that combine reconstruction error with supervised or semi-supervised decision mechanisms may further enhance robustness in complex cyber–physical environments.

From a deployment perspective, the relatively compact architecture and stable inference behavior indicate that the model is suitable for IoT-enabled engineering systems with constrained resources. Future work may therefore explore edge-assisted or hierarchical deployment scenarios, where real-time anomaly detection is required under limitations in latency, bandwidth, and computational capacity. This balance between detection performance and efficiency is critical for large-scale smart infrastructure applications.

Furthermore, a more comprehensive evaluation is required to strengthen the validity of the approach. This includes benchmarking against alternative methods such as LSTM autoencoders, CNN autoencoders, Isolation Forest, and statistical anomaly detection techniques. In addition, incorporating precision–recall AUC (PR-AUC) is recommended to provide a more informative evaluation under highly imbalanced conditions.

5. Conclusion

This study proposed a hybrid CNN–LSTM autoencoder framework for anomaly detection in IoT-enabled smart engineering systems using multivariate time-series data. The unsupervised approach effectively learned normal system behavior by capturing both spatial and temporal dependencies, demonstrating stable convergence and good generalization on the BATADAL-A cyber–physical dataset. Experimental results show strong performance in identifying normal operational states, achieving 90% overall accuracy with low false alarm rates. However, anomaly detection performance remains limited due to low recall and F1-score for anomalous events, reflecting the impact of class imbalance and the inherent limitations of reconstruction-based methods. The model achieved an AUC of 0.677, indicating moderate discriminative capability between normal and abnormal behavior.

Although the framework is suitable for reliable monitoring in IoT-enabled engineering systems, further improvements are required to enhance detection sensitivity for rare and safety-critical anomalies. Future work will focus on imbalance-aware learning strategies, adaptive thresholding mechanisms, explainable AI integration, federated learning approaches, and comparisons with transformer-based anomaly detection models.

6. Declarations

6.1. Author Contributions

Godfrey Perfectson Oise: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – Original Draft, Visualization; **Kevin Chinedu Pius:** Data Curation, Validation, Formal analysis, Investigation, Writing – Review & Editing; **Felix Oshiorenoya Uloko:** Resources, Validation, Investigation, Writing – Review & Editing; **Immunhierokene Clinton Obrorindo**: Supervision, Project administration, Resources, Writing – Review & Editing; **Roli Lydia Oshasha:** Visualization, Data Curation, Supervision, Writing – Review & Editing.

6.2. Institutional Review Board Statement

Not applicable.

6.3. Informed Consent Statement

Not applicable.

6.4. Data Availability Statement

The data presented in this study are available at: https://www.kaggle.com/datasets/minhbtnguyen/batadal-a-dataset-for-cyber-attack-detection/data.

6.5. Acknowledgment

Not applicable.

6.6. Conflicts of Interest

The authors declare no conflict of interest.

7. References

[1] N. Omer, A. H. Samak, A. I. Taloba, and R. M. Abd El-Aziz, "A novel optimized probabilistic neural network approach for intrusion detection and categorization," *Alexandria Engineering Journal*, vol. 72, pp. 351–361, Jun. 2023. https://doi.org/10.1016/j.aej.2023.03.093.

[2] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," *Future Generation Computer Systems*, vol. 133, pp. 95–113, Aug. 2022. https://doi.org/10.1016/j.future.2022.03.001.

[3] A. Abusitta, G. H. S. de Carvalho, O. A. Wahab, T. Halabi, B. C. M. Fung, and S. Al Mamoori, "Deep learning-enabled anomaly detection for IoT systems," *Internet of Things*, vol. 21, p. 100656, Apr. 2023. https://doi.org/10.1016/J.IOT.2022.100656.

[4] S. H. Rafique, A. Abdallah, N. S. Musa, and T. Murugan, "Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection—Current Research Trends," *Sensors*, vol. 24, no. 6, Mar. 2024. https://doi.org/10.3390/S24061968.

[5] J. Nayak, B. Naik, P. B. Dash, S. Vimal, and S. Kadry, "Hybrid Bayesian optimization hypertuned catboost approach for malicious access and anomaly detection in IoT nomalyframework," *Sustainable Computing: Informatics and Systems*, vol. 36, Dec. 2022. https://doi.org/10.1016/j.suscom.2022.100805.

[6] G. P. Oise *et al.*, "Decentralized Deep Learning in Healthcare: Addressing Data Privacy with Federated Learning," *FUDMA Journal or Sciences*, vol. 9, no. 6, pp. 19–26, Jun. 2025. https://doi.org/10.33003/fjs-2025-0906-3714.

[7] S. A. Oyedotun, G. P. Oise, and C. E. Ozobialu, "Towards Intelligent Cybersecurity in SCADA and DCS Environments: Anomaly Detection Using Multimodal Deep Learning and Explainable AI," *Journal of Science Research and Reviews*, vol. 2, no. 3, pp. 20–31, Jul. 2025. https://doi.org/10.70882/josrar.2025.v2i3.76.

[8] G. P. Oise, O. C. Nwabuokei, O. J. Akpowehbve, B. A. Eyitemi, and N. B. Unuigbokhai, "Towards Smarter Cyber Defense: Leveraging Deep Learning for Threat Identification and Prevention," *FUDMA Journal or Sciences*, vol. 9, no. 3, pp. 122–128, Mar. 2025. https://doi.org/10.33003/fjs-2025-0903-3264.

[9] Y. Kayode Saheed, O. Harazeem Abdulganiyu, and T. Ait Tchakoucht, "A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 5, May 2023. https://doi.org/10.1016/j.jksuci.2023.03.010.

[10] S. M. Rajagopal, M. Supriya, and R. Buyya, "FedSDM: Federated learning based smart decision making module for ECG data in IoT integrated Edge–Fog–Cloud computing environments," *Internet of Things (Netherlands)*, vol. 22, Jul. 2023. https://doi.org/10.1016/j.iot.2023.100784.

[11] M. M. Inuwa and R. Das, "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks," *Internet of Things (Netherlands)*, vol. 26, Jul. 2024. https://doi.org/10.1016/j.iot.2024.101162.

[12] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered Intrusion Detection System," *Internet of Things (Netherlands)*, vol. 24, Dec. 2023. https://doi.org/10.1016/j.iot.2023.100936.

[13] G. Oise and S. Konyeha, "Environmental impacts in e-waste management using deep learning," *Discover Artificial Intelligence*, vol. 5, no. 1, p. 210, Aug. 2025. https://doi.org/10.1007/s44163-025-00376-9.

[14] A. Abusitta, M. Q. Li, and B. C. M. Fung, "Malware classification and composition analysis: A survey of recent developments," *Journal of Information Security and Applications*, vol. 59, Jun. 2021. https://doi.org/10.1016/j.jisa.2021.102828.

[15] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Comput. Secur.*, vol. 94, Jul. 2020. https://doi.org/10.1016/j.cose.2020.101863.

[16] E. Tsogbaatar *et al.*, "DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT," *Internet of Things (Netherlands)*, vol. 14, Jun. 2021. https://doi.org/10.1016/j.iot.2021.100391.

[17] J. Jiang *et al.*, "A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams," *Comput. Commun.*, vol. 194, pp. 250–257, Oct. 2022. https://doi.org/10.1016/j.comcom.2022.07.034.

[18] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, Jun. 2020. https://doi.org/10.1016/j.future.2020.02.017.

[19] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for Industrial IOT environment," *Expert Syst. Appl.*, vol. 249, Sep. 2024. https://doi.org/10.1016/j.eswa.2024.123808.

[20] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 167–185, Jan. 2024. https://doi.org/10.1016/j.iotcps.2023.12.003.

[21] Minh T. Nguyen, "BATADAL: Cyber Attacks Detection in Water Systems," 2023, *https://www.kaggle.com/datasets/minhbtnguyen/batadal-a-dataset-for-cyber-attack-detection/data*.

[22] T. C. Doan, Q. T. Duong, T. H. H. Nguyen, and V. D. Pham, "AI-Driven Anomaly Detection in IoT Systems: Techniques and Applications," in *Innovations and Challenges in Computing, Games, and Data Science*, IGI Global Scientific Publishing, 2025, pp. 29–44. https://doi.org/10.4018/979-8-3373-2647-4.ch002.

[23] G. P. Oise, "E-ViTNet: A lightweight vision transformer with oppositional cat swarm optimization for automated E-Waste sorting," *Next Research*, vol. 6, p. 101373, Apr. 2026. https://doi.org/10.1016/j.nexres.2026.101373.

[24] R. Atassi, "Anomaly Detection in IoT Networks: Machine Learning Approaches for Intrusion Detection," *Fusion: Practice and Applications*, vol. 13, no. 1, pp. 126–134, 2023. https://doi.org/10.54216/FPA.130110.

[25] M. A. Alsoufi *et al.*, "Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review," *Applied Sciences*, vol. 11, no. 18, p. 8383, Sep. 2021. https://doi.org/10.3390/app11188383.

[26] A. Abusitta, T. Halabi, A. S. Bataineh, and M. Zulkernine, "Generative Adversarial Networks for Robust Anomaly Detection in Noisy IoT Environments," in *ICC 2024 - IEEE International Conference on Communications*, IEEE, Jun. 2024, pp. 4644–4649. https://doi.org/10.1109/ICC51166.2024.10622882.