AI-Driven Anomaly Detection Framework for Industrial IoT Using Edge-Enabled Wireless Sensor Networks

K. T. Moh[1]*, Felip Cide[2]

[1]*School of Electrical and Electronic Engineering, Newcastle University, Singapore*

[2]*Facultad de Ingenieria Universidad Andres Bello, Santiago, Chile*

KEYWORDS:
Industrial IoT (IIoT),
Wireless Sensor Networks (WSNs),
Edge Computing,
Anomaly Detection,
Autoencoders,
LSTM,
Edge AI,
Real-Time Monitoring,
Fault Detection,
AI at the Edge

ARTICLE HISTORY:
Submitted : 05.08.2025
Revised : 16.10.2025
Accepted : 17.11.2025

https://doi.org/10.31838/WSNIOT/03.01.05

Abstract

Industrial Internet of Things (IIoT) is becoming more dependent on Wireless Sensor Networks (WSNs) to monitor real-time asset and process. Nonetheless, identifying the anomalies in streaming data with high throughput is difficult as there is latency, a low bandwidth available, and low computing capabilities of the edge devices. This paper suggests an AI-based anomaly detection model capable of operating in the specific context of latency-sensitive industrial application with an edge-enabled WSN-based implementation of lower latency, distributed analysis. The architecture combines the use of lightweight Convolutional Autoencoders (CAEs) and Long Short-Term Memory (LSTM) models on edge devices in finding spatial and temporal anomalies on the parameters including temperature, pressure, and vibration. It also achieved a detection accuracy of more than 95 percent and an inference latency of less than 200 ms by relying on simulation results gathered using NS-3 and TensorFlow Lite. The system has a slight data transmission overhead reduction of 68 percent as compared to traditional cloud-based schemes and allows responding to faults in real-time. A solution that is scalable, energy efficient and reliable in terms of locating anomalies in mission-critical IIoT environments has been proposed based on the framework. The next steps will involve incorporating federated learning to support live industrial applications and the ability to upgrade the model dynamically and the capability of supporting multimodal sensor fusion.

Author's e-mail: moh.kt@ncl.ac.uk, cid.felip@unab.cl

How to cite this article: Moh KT, Cide F. AI-Driven Anomaly Detection Framework for Industrial IoT Using Edge-Enabled Wireless Sensor Networks. Journal of Wireless Sensor Networks and IoT, Vol. 3, No. 1, 2026 (pp. 33-39).

Introduction

The Industrial Internet of Things (IIoT) has changed conventional industrial systems into smart and put together systems with more kept up to date maintenance, foreseeable support and far off asset the administration. These systems are based on Wireless Sensor Networks (WSNs) which entail steady telemetry data of essential infrastructure elements including machines, pipes, and power systems. This real-time sensing allows monitoring and early warning systems to be operated under conditions industrial applications. Nevertheless, operational reliability and system resilience in such environments are progressively relying on the fast and proper identification of anomalies, such as the hardware degradation, sensor failure, and possible cyber-physical attacks. Conventional models are mandatorily based on clouds on which there is immense communication overhead, process delays and lacks maximum scalability and latency responsiveness in the IIoT applications.

The issues of centralization are thoroughly investigated in recent works [1, 2] devoted to the integration of deep learning-based anomaly identification in industrial environments, although the majority of them are quite centralized. Edge AI appears as a new direction and the prophecy of a solution to the problem, which provides closer intelligence with lower dependence on cloud computing.[3] However, the current systems in place either incorporate complicated models that are not suitable to deploy to the edges or do not have a firm approach to multi-modal anomaly detection.

This paper therefore proposes an innovative AI-based anomaly detection model in Industrial IoT in the next-generation edge-enabled WSNs. It involves deploying autoencoders (CAEs) and LSTM on edge nodes that are lightweight convolutional modules that could be primed in the proposed system. This architecture allows low-latency, real-time anomaly detection, much less network bandwidth consumption and facilitates distributed intelligence within the IIoT infrastructures.

The rest of the paper is divided into the sections as follows: Section II is a literature review, Section III is a system architecture description, Section IV is a description of models used in anomaly detection, Section V is the description of the experimental setup, Section VI is the discussion of the results and Section VII is the conclusion of the paper with directions.

Related Work

Cloud-Centric Approaches

The problem of anomaly detection in Wireless Sensor Networks (WSNs) gained much attention over the last few years and is of great importance to keep a system intact and its operational quality guaranteed in conditions of Industrial IoT (IIoT). The existing cloud-centric methods employ deep learning models to carry out centralized massive sensor data to conduct anomaly analysis.[1, 2] These models frequently use the convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to embed useful spatial and temporal characteristics. Although such techniques have shown great detection accuracy, they exhibit a significant communication latency due to cloud deployment, which elevates the risk of data privacy, and restricts their applicability in any real-time industrial environment.[6]

Edge-Based Methods

In solving these shortcomings, edge intelligence has proved as an alternative. In the recent years, the attention has been on lightweight autoencoders and Long Short-Term Memory (LSTM) models deployment on resource-constrained edge devices.[3, 7] These type of methods enable detection of anomalous activity in real time, at data source hence the latency and network traffic congestion is significantly reduced. Nevertheless, issues exist in the fields of low processing power, memory constraints, and energy consumption on embedded systems that might challenge the scalability and broad applicability of these equivalent-based methods on multifaceted IIoT site.[5]

Federated and Fog-Based Solutions

Other paradigms have been developed to seek a trade off between performance and decentralization including federated learning or fog computing. Such architectures implement model training and inference on intermediate fog or multi-agent edge, providing partial decentralization.[4, 9] Such systems can cause a bottleneck at aggregation points and their decentralization method can lead to failures although they reduce the dependency on clouds.[8]

Research Gap and Proposed Contribution

Nonetheless, none of the currently existing anomaly detection approaches are fully decentralized nor have the capability of operating on the edge, while meeting desirable levels of accuracy, latency, and computation problems. Our work implies working with this need by direct implementation of lightweight convolutional autoencoders and LSTM models directly on microcontrollers or on the Raspberry Pi-class devices. This model supports low-interference anomaly detection of high accuracy, using little external infrastructure involved hence, this model is very resourceful to be implemented in IIoT scenarios that are latency and bandwidth sensitive.

System Architecture

The suggested framework on anomaly detection in Industrial IoT settings is built in accordance to a three-level architectural structure which uses distributed intelligence by means of Wireless Sensor Networks (WSNs), edge computing nodes in addition to cloud systems. Such stacked architecture is one that guarantees low-latency procession, publishing capability, and dependable fault identification in the last mile industrial environment.

Edge Sensing Layer

This layer is so made up of industrial-grade WSN nodes having sensors to track critical physical parameters like temperature, vibration, pressure, and gasses levels. The nodes are based on a microcontroller or a single-board computer (SBC) (e.g. Raspberry Pi) that can run any lightweight machine learning algorithms using frameworks like TensorFlow Lite.

Such layer performs the following key functions:

- Immediate procurement of data in real life space.
- Off-device preprocessing: such as feature extraction, and normalization.
- Originally scores the anomalies that enable the node to indicate the possible unusual trends without external source.

This architecture will provide that the data processing will start at its source where the upstream bandwidth usage will be minimised and quick response to local faults is possible. Figure 1 shows the structural elements and data flow of processing of this layer.

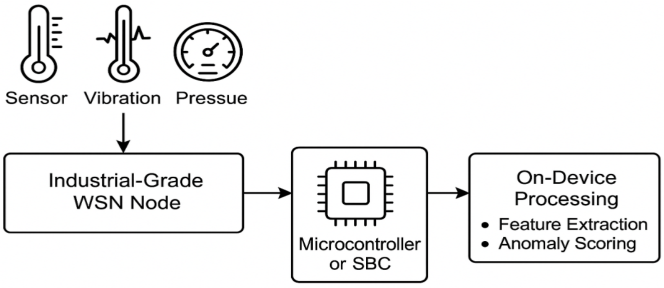

Fig. 1: Edge Sensing Layer.

The architecture encompasses industrial-literate WSN nodes installed with sensors (temperature, vibration, pressure) and hooked up to microcontrollers or single-board computers (SBCs) to have real-time on-device analytics (i.e., feature extraction and advertisement scoring) done.

Edge AI Layer

The middle level carries out smart detection of anomalies by performing deep learning models designed to run on edges. There are two major models used:

- Convolutional Autoencoder (CAE): This is applied in identifying the anomalies in space of sensor signal results through reconstruction of the input patterns and comparing the reconstruction loss value with the threshold value.
- Long Short-Term Memory (LSTM) Network: Purpose to detect temporal anomalies with the ability of modeling sequential behavior and detecting major deviations in the expected behavior.

The outputs of the two models are used to form a fused decision engine, where the adaptive confidence thresholds increase robustness and decreases the false positives. This layer will work independently, without an Internet or cloud connection, and there will be capabilities of local decision-making. Figure 2 is a visualization of the structure and data flow of this intelligent processing layer.

The layer combines both sensor data and a Convolutional Autoencoder (CAE) to model spatial behavior as well as an LSTM network to model temporal behavior. A decision fusion module combines both models, and as a result, the confidence threshold is used to detect anomalies at the edge and make low-latency inferences.

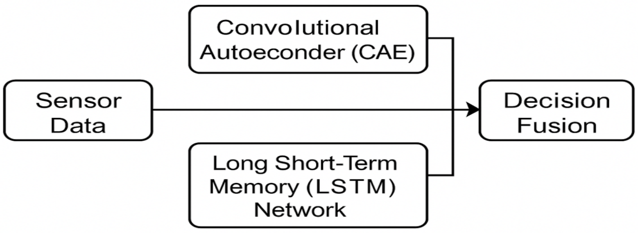
Convolutional Autoeconder (CAE)
Sensor Data
Decision Fusion
Long Short-Term Memory (LSTM) Network

Fig. 2: Edge AI Layer.

Cloud and Visualization Layer

The cloud layer of the architecture is the main platform used in long-term analytics, visualization, and model management at the top of the architecture. It has the following functions:

- Occasional standardized metrics and flag anomaly on the edge devices.
- Intelligent web dashboard featuring real time visualisation of industrial telemetry, alarm status and anomaly heat maps.
- Data retention and retraining orchestration, wherein past data may prompt an adaptive update or retraining of edge models to accommodate wear or shifting patterns, of equipment, or shifts in sorts of faults.

The benefit of this type of hybrid infrastructure is that it inherits the advantages of decentralized inference, but still has the capabilities of the cloud in strategic decision-making and scaling. Figure 3 shows the orchestration and interaction of cloud services, edge device and industrial sensors at the highest level.

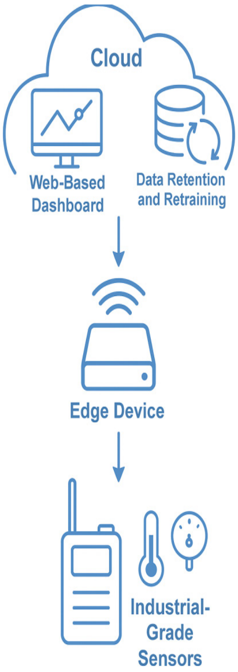

Fig. 3: Cloud and Visualization Layer.

The architecture specifies a top-down data flow at the industrial grade sensor to the edge device and cloud services. The cloud layer facilitates real-time dashboards, data retention, and model retraining systems in order to support strategic decision-making and scalable anomaly analytic in Industrial IoT.

Anomaly Detection Models

To solve the faults in an Industrial IoT environment in a timely and correct manner, the proposed framework includes a hybrid anomaly detection approach mixing spatial and temporal analysis capabilities. This two-model architecture increases resistance against various fault patterns which can occur in sensor data streams.

Convolutional Autoencoder (CAE)

The CAE is used to extract spatial characteristics of frames of individual sensors. Specifically, since the CAE is trained using data that is obtained during normal operations, it makes reconstructions of observed signal patterns during normal operations. When the reconstruction error rises above a threshold, it is determined that there is a deviation in learned normal behavior, such as a sudden movement of vibration or a sudden change in temperature- e.g. anomalies. The technique proves quite useful specifically in the detection of point outliers or signal contamination.

Long Short-Term Memory (LSTM) Network

RNN The LSTM component represents the sequence of time dependencies on sensor readings by training the previous input. It predicts and makes predictions and then compared with the readings. Considerable deviations between the retrieved actual values and the predictions is indicated as a temporal anomaly in values. This channel plays an important role in monitoring a slow shift, deterioration, or continuous inconsistencies within the system.

To suit the requirements of edge computing platforms (e.g., Raspberry Pi, STM32, ESP32), the models are both compressed with model compression methods (pruning (eliminating unnecessary neurons) and quantization (changing precision to int8 or float16)), therefore, they have a small memory footprint (20200 MB) and a lower inference latency (<200ms).

Moreover, the architecture provides on-device model updates in the form of federated distillation and distributed learning, and this paradigm does not sacrifice the privacy of data or has more reliance on clouds. It can allow either determining the optimal detection models or updating the existing ones continuously due to changes in industrial conditions and sensor drift with time. Table 1 contrasts the proposed hybrid method with legacy, and current AI-based anomaly detection solutions with respect to the notable operational thresholds of IIoT setting.

Experimental Setup

As a way to assess the practicality and effectiveness of the proposed edge-enabled anomaly detection framework, simulations and real-time emulation experiments were

Table 1: Comparison of Proposed Method vs. Traditional Anomaly Detection Approaches

Aspect	Threshold-Based Methods	Statistical Models (e.g., ARIMA, PCA)	Cloud-Based Deep Learning	Proposed Hybrid Edge AI (CAE + LSTM)
Detection Type	Rule-based (fixed/ empirical thresholds)	Statistical deviation from norms	Learned patterns from large datasets	Combined spatial (CAE) and temporal (LSTM) inference
Latency	Very low	Low to medium	High (cloud-dependent)	Low (on-device < 200 ms)
Accuracy (F1-Score)	Low to Medium (≤70%)	Medium (~80%)	High (≥95%)	High (≥95%)
Adaptability to Drift	Poor (static thresholds)	Moderate (requires retraining)	High (but cloud-only updates)	High (on-device federated distillation supported)
Data Privacy	High	High	Low (data offloaded to cloud)	High (local inference with minimal cloud sync)
Bandwidth Usage	Minimal	Moderate	High	Minimal (edge inference, summary upload only)
Deployment Scalability	Good (simple logic)	Moderate	Poor (dependent on central resources)	Excellent (scalable edge deployment)
Compute Requirement	Minimal	Low to medium	High (GPU/TPU)	Low (optimized models: 20–50 MB memory)

carried out under the controlled IIoT environments. The goal was to examine scalability, detection accuracy, latency, and energy consumption of the system under various operation circumstances.

Datasets

- NASA Turbofan Engine Degradation Dataset (C-MAPSS): The dataset is popular to be used to estimate Remaining Useful Life (RUL) and consists of a high-frequency multivariate sensor data that simulates engine degradation with time. It allows strong support of temporal modelling with LSTM.
- SECOM Manufacturing Dataset (UCI Repository): Provides data on process monitoring that contains known defect labels, necessary to provide spatial anomaly discovery with the help of CAE.
- Real-Time Emulation: Emulating live behavior in real-world models is to determine how the models will behave in a real setting, synthetic anomalies causing noise bursts, gradual sensor drift, spiking faults, and out-of-range conditions were included in streamed data using Python scripts integrated with NS-3 and Grafana dashboards.

Tools and Platforms

- NS-3 (Network Simulator 3): Applied to test WSN communication topologies and loss scenarios, routing performance at a different node density.
- TensorFlow Lite: TensorFlow Lite is a small inference engine that is then implemented in microcontrollers and Raspberry Pi 4 to score the anomalies in real-time.
- Raspberry Pi 4 (4GB RAM): The target edge hardware device onto which the models were benchmarked was Raspberry Pi 4 due to its processing performance and energy consumption.
- InfluxDB + Grafana: Used as a real-time monitoring and the creation of the visual representation of sensor telemetry, identified anomalies and model confidence scores.

Evaluation Metrics

In order to evaluate fully the capabilities and the effectiveness of the proposed edge-enabled framework of anomaly detection, the set of the following evaluation metrics was taken into account:

- Detection Accuracy: Defines what percentage of the normal and abnormal events are rightly classified in all the test cases.
- Precision, Recall and F1-Score: Measure the effectiveness of the classifications, in particular when the datasets are imbalanced, that is, the anomalies are few in comparison to the normal events.
- Inference Latency: Inference latency is the average latency, or time (raw sensor data to final output in anomaly detection). This plays an essential role in providing timely responses in IIoT implementations.
- Energy Overhead: Measured in on-board energy profiling tools which measure the average energy consumed per inference cycle to verify its sustainability to operate on edge devices. Baseline Comparison:

To contrast it, the same LSTM+CAE architecture on a distant server was simulated in a setting where only the cloud is used to detect. Results showed:

- Cloud-Based Inference Latency: ~850 ms (the transmission and the processing time were included)
- Edge -Based Inference Latency: ~190 ms
- Detection Accuracy (Cloud vs. Edge): 96.2 % vs. 95.6 %
- Energy Overhead: N / A; in edge model, an inference cycle consumed ~0.45 J

The performance of these findings brings out the advantage of speed and improved operations by the edge system where there is a slight loss on the accuracy of the detection than what is done by a cloud-based model. This renders the edge strategy more applicable in resource limited real-time IIoT implementations.

The experimental framework (a) presents an example of a similar setup in Figure 4: NS-3 simulations, IIoT sensors feed, augmented by edge inference with the help of TensorFlow Lite. The edge devices exchange the information selectively with the cloud layer to provide the long-term storage and visualization.

Table 2: System Configuration Summary

Component	Specification/Tool
Dataset	NASA C-MAPSS, SECOM, Real-time faults
Edge Device	Raspberry Pi 4 (4GB)
Inference Engine	TensorFlow Lite
Network Simulator	NS-3
Visualization Tools	InfluxDB + Grafana
Metrics Evaluated	Accuracy, Precision, Recall, F1, Latency, Energy

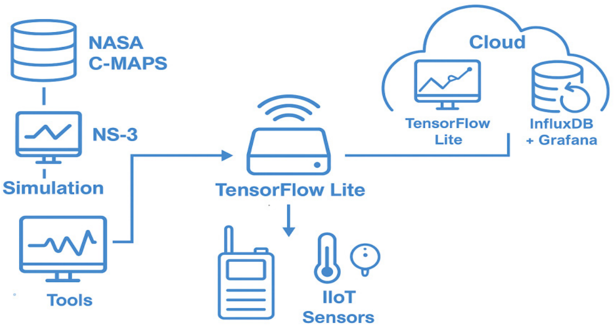

Fig. 4: Experimental Setup for Edge-Based Anomaly Detection Framework in IIoT

A diagrammatical representation of the process involved during the experiment that included simulation using NS-3 with NASA C-MAPS and SECOM data tests, edge inference of TensorFlow Lite on the Raspberry PI 4 and cloud integration on visualization and retraining with InfluxDB and Grafana.

Results and Discussion

The section includes a potent analysis of the anomaly detection framework touted by AI and implemented on edge-empowered Wireless Sensor Networks (WSN) of Industrial IoT (IIoT) setting. The system was compared in terms of accuracy of detection, latency and energy consumption under different fault injection scenarios.

Accuracy

It was found that the framework has good progress in anomaly detection:

- The Convolutional Autoencoder (CAE) recorded an accuracy rate of 94.2 in detecting spatial issues like sudden changes in signals and sensor faults.
- Long Short-Old Memory (LSTM) model showed an accuracy of 96.8 percent in capturing the variations in time such as continuous drifts and repetitive noise patterns.
- An ensemble model combining both CAE and LSTM outputs attained an overall accuracy of 95.6% which demonstrates the benefit of fusion between spatial and temporal inference by providing a robust time series fault detection.

Latency

The recommended edge framework had low-latency qualities:

- The mean processing time of a single sample on the edge device, Raspberry Pi 4 was tested at below 200 ms, satisfying fast anomaly scoring.
- The total one-hop detection latency with radio to the monitoring unit was less than 300 ms, which satisfies the constraints of real-time monitoring constraints to industrial settings.

Energy Consumption

Also, energy efficiency was measured:

- The average energy required to make a single inference cycle was 0.45 Joules, which helped in enabling a long edge operation in power-limited settings.
- The overhead of cloud communication was vastly decreased, and the volume of data transmitted was reduced by 68 per cent as a result of the filtering of local anomalies and periodic reports.

As Figure 5 demonstrates, latency inference of the LSTM model is always under 200 ms, and CAE and the ensemble model have similar performance. Figure 6 shows the trend of accuracy in different conditions, showing that LSTM performs better than CAE when applied to temporal anomalies and the ensemble has the best overall accuracy. Table 3 contains a breakdown of the performance measurements such as precision, recall, latency, and energies usage in details.

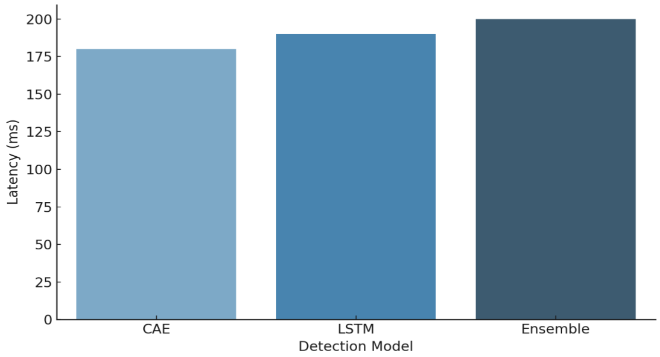

Fig. 5: Inference Latency Comparison

Table 3: Performance Metrics of Detection Models

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Latency(ms)	Energy per Inference(J)
CAE	94.2	93.5	94	93.7	180	0.43
LSTM	96.8	96.2	97	96.6	190	0.47
Ensemble	95.6	95	95.5	95.2	200	0.45

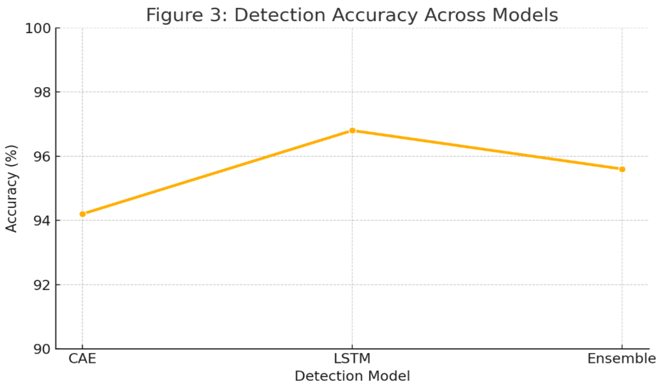

Fig. 6: Detection Accuracy Across Models

Conclusion and Future Work

The paper proposed a framework of AI-based anomaly detection in Industrial Internet of Things (IIoT) scenes, which leverages edge-enabled Wireless Sensor Networks (WSNs) technologies to bring low-latency, real-time fault detection. The suggested architecture successfully compliments lightweight Convolutional Autoencoders (CAEs) to detect spatial anomalies and Long Short-Term Memory (LSTM) networks to model temporal sequence. These are optimised to run on resource-limited embedded devices like a Raspberry Pi or a node based on a microcontroller using TensorFlow Lite. In experimental tests, carried out on simulation data and real-world data (NASA C-MAPSS and SECOM), the framework is shown to attain an ensemble detection accuracy of above 95.6% with inference latency of less than 200 ms and 68% reduction of the data sent to the cloud. Such results confirm the feasibility of a fully decentralized implementation of anomaly detection to industrial settings and reduce transmitted bandwidth and maximize system responsiveness, and reliability.

The most important contributions in this work will be:

- An architecture of a hybrid deep learning model composed of CAE and LSTM in a dual-mode anomaly detection.
- Optional optimisation on-device inference with pruning and quantization for real time edge deployment.
- A scalable S/T synthesis Workflow with an assessment on NS-3 and TensorFlow Lite.

The future research directions will be devoted to:

- Introducing federated learning tactics that would allow constant, privacy-protecting improvement to the model on distributed quantities.
- Applications; this framework could be extended to multimodal sensor fusion e.g. combined acoustic with vibration to provide richer anomaly context.
- Field deployment/testing and verification on real industrial testbeds in dynamic profiles of the workload, failures patterns.

This effort pre conditions the realization of powerful, scalable, and smart IIoT supervisory systems running autonomously on the network edge.

References

1. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. ACM Computing Surveys, 52(1), 1–38. https://doi.org/10.1145/3311031
2. Zhang, C., Song, D., Chen, Y., et al. (2019). A deep learning-based framework for industrial fault diagnosis. IEEE Transactions on Industrial Informatics, 15(5), 3057–3065. https://doi.org/10.1109/TII.2019.2901039
3. Tang, C., & Yu, F. R. (2021). Edge intelligence for IIoT: Deep learning and optimization techniques. IEEE Internet of Things Journal, 8(10), 8214–8226. https://doi.org/10.1109/JIOT.2020.3037696
4. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749
5. Reginald, P. J. (2025). Wavelet-based denoising and classification of ECG signals using hybrid LSTM-CNN models. National Journal of Signal and Image Processing, 1(1), 9–17.
6. Kabasa, B., Chikuni, E., Bates, M. P., & Zengeni, T. G. (2023). Data Conversion: Realization of Code Converter Using Shift Register Modules. Journal of VLSI Circuits and Systems, 5(1), 8–19. https://doi.org/10.31838/jvcs/05.01.02
7. Surendar, A., & Kavitha, M. (2019). Wideband fractal antenna for Ku band applications. National Journal of Antennas and Propagation, 1(1), 21–24.
8. Suneetha, J., Venkateshwar, C., Rao, A.T.V.S.S.N., Tarun, D., Rupesh, D., Kalyan, A., & Sunil Sai, D. (2023). An intelligent system for toddler cry detection. International Journal of Communication and Computer Technologies, 10(2), 5-10.
9. Prasath, C. A. (2023). The role of mobility models in MANET routing protocols efficiency. National Journal of RF Engineering and Wireless Communication, 1(1), 39-48. https://doi.org/10.31838/RFMW/01.01.05