JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 278-294
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Hybrid Edge–Cloud Deep Learning Framework for Real-Time Anomaly Detection in Industrial IoT Networks

Mohammad Nasar[1]*, Mohammad Abu Kausar[2]*

[1] *Computing and Informatics Department, Mazoon College, Muscat, 133, OMAN*

[2] *Department of Information System, University of Nizwa. Nizwa, OMAN*

*Corresponding Author: nasar31786@gmail.com, kausar4u@gmail.com
DOI: https://doi.org/10.30880/jscdm.2026.07.02.016

Article Info

Received: 10 October 2025
Accepted: 6 March 2026
Available online: 30 June 2026

Keywords

Industrial Internet of Things (IIoT), anomaly detection, edge computing, cloud computing, deep learning, hybrid framework, real-time security, federated learning (FL), CNN–LSTM, Edge-IIoTset dataset

Abstract

Industrial Internet of Things (IIoT) networks produce huge amounts of diverse data from different sensors, actuators, and smart objects distributed in mission-critical industrial environments. Maintaining the safety and trustworthiness of these systems requires real-time anomaly detection mechanisms that can identify malicious operations, malfunctioning devices, or abnormal traffic patterns with high accuracy and low latency. Cloud-based deep learning solutions provide computational scalability but may introduce significant communication delays. Although edge solutions deliver proximity and fast inference, they lack sufficient computational capability to handle complex deep models. Motivated by this gap in the existing literature, in this study, we introduce a novel hybrid edge–cloud deep learning framework for real-time anomaly detection in IIoT networks that capitalizes on the strengths of both paradigms. The edge layer conducts lightweight preprocessing, feature extraction, and initial detection with the help of compressed neural modules, while the cloud layer runs advanced deep models (e.g., CNN–LSTM hybrids and transformer-based ensembles) to carry out high-level analysis. A joint inference pipeline minimizes the end-to-end delay with adaptive offloading strategies modelled as an optimization problem that trades off detection accuracy for network delay. Extensive experiments on the Edge-IIoTset dataset show that the proposed hybrid architecture achieves a detection accuracy of 98.7% with a 37% average inference latency reduction compared to purely cloud-based solutions. The comparison also demonstrates improved robustness across diverse attack categories. These results also demonstrate the feasibility of hybrid edge–cloud intelligence for secure, robust, and real-time anomaly detection in IIoT.

1. Introduction

The Internet of Things (IoT) has developed into an Industrial Internet of Things (IIoT) for Industry 4.0, reshaping the development of factories, smart grids, and critical infrastructures to consist of billions of heterogeneous objects in a position to communicate and interoperate intelligently [1]. Such systems share large amounts of time-series sensor data, control commands, and messages between sensors to accomplish tasks as diverse as predictive

maintenance, safety surveillance, or process automation [1], [2]. However, the high connectivity of the IIoT makes it susceptible to various cybersecurity and operational anomalies. Fake data injection, distributed denial-of-service (DDoS) attacks, malware propagation, and insider misuse may disrupt normal industry processes, resulting in financial losses and even threatening human lives [3], [4]. However, such anomaly detection (requiring the detection of deviations from normal trends in traffic and operation) has become an essential requirement for securing IIoT infrastructures because a layered approach is generally employed here.

Conventional anomaly detection systems using static rules, thresholds on specific metrics, or simple machine learning classifiers are not very practical for IIoT. They suffer from high false positive rates, poor portability, and difficulty in identifying new attack vectors over time in heterogeneous data environments [5], [6]. However, deep learning (DL) has recently displayed promising effectiveness in capturing the nonlinear temporal–spatial dependencies and extracting representative features from raw IoT traffic [7]. Various types of architectures, including convolutional neural networks (CNNs) [5], [6], recurrent neural networks (RNNs), autoencoders, and transformer-based models, have been used to improve anomaly detection accuracy [8], [9]. For example, CNNs can effectively handle spatial patterns in traffic features, and Long Short-Term Memory (LSTM) models are strong for exploring the temporal correlations of sensor sequences. Meanwhile, autoencoders can compact high-dimensional traffic into latent spaces and emphasize abnormal behaviors via reconstruction errors [10].

However, applying DL methods in practical IIoT systems faces many challenges. Cloud-based solutions, in which industrial devices send data to a remote server for centralized processing, have scalability and computational resources. The cloud allows the training and deployment of complex DL models, such as hybrid CNN–LSTM or transformer ensembles, which achieve state-of-the-art anomaly detection performance [12]. However, cloud-based techniques have several disadvantages. First, large amounts of data being transmitted through a network tend to suffer from communication delay and congestion [14], which could be intolerable in mission-critical scenarios where every millisecond counts for the safety of humans. Second, the storage and processing of sensitive industrial data at a centralized level would compromise privacy and introduce a single point of failure [13].

Edge computing offers an alternative by performing computational tasks locally at or near the IIoT devices. In this paradigm, applications can make real-time decisions without high latency costs without having to upload and store big data sets or even run operations with reliable connectivity to cloud [3], [14]. Edge deployment of lightweight DL models for fast anomaly detection in applications such as robotic assembly lines and smart grid substation monitoring. However, the computing power, memory, and energy capability of edge devices are far from sufficient to execute resource-demanding DL models [8]. In the case of the edge, these solutions generally present a compromise between the previously mentioned accuracy of detection and speed of response.

This trade-off between latency and accuracy underpins the emergence of hybrid edge–cloud architectures, which aim to integrate the analytical depth of cloud systems with the real-time responsiveness of edge computing. These architectures correctly leverage the capability for anomaly detection at the edge and cloud, thus allowing the benefits of both [4], [15]. In this paradigm, the edge performs the initial pre-processing, feature transformation, and light anomaly scoring pipeline, while the cloud oversees complex DL pipelines (e.g., deep ensemble) and global-scale correlation. This collaborative framework improves scalability, maintains real-time responsiveness, and reduces the communication load [9], [16]. It also follows the recent trend of privacy-preserving computations, in which raw data are kept at the edge and only derived features or abnormality indications are shared with the cloud for further processing [18].

However, there are some limitations to the present study. In general, hybrid solutions are focused on architectural design and mostly do not involve a formal heuristic for the latency-accuracy trade-off. For example, to the best of our knowledge, few studies have clearly cast anomaly detection as a constrained optimization problem that minimizes end-to-end latency under accuracy constraints [11], [21]. Several prior studies have also modelled edge and cloud intelligence as separate modules rather than coordinated pipelines while using the initial edge inference to speed up further analysis at the system level [8], [22]. Another difficulty is the shortage of evaluations on practical large-scale datasets. Synthetic data have been widely used as ground truth in research; however, they cannot model the complexities of real-world IIoT scenarios, including mixed traffic sources from industrial protocol-encrypted flows and the diverse behavior of devices. The Edge-IIoTset dataset [24] is available, and it includes both IoT and IIoT realistic traffic for different attack scenarios, allowing more suitable evaluations to be performed.

In this study, we address these challenges by presenting a Holistic Edge–Cloud Deep Learning Framework for Real-Time Anomaly Detection in IIoT Networks. The architecture adds a two-stage inference process, namely, light pre-processing and anomaly rating at the edge, with compressed CNN–autoencoder models, supplemented by enhanced analysis in the cloud, where hybrid CNN–LSTM–Transformer ensembles are deployed. Notably, by explicitly modelling the workflow as an optimization problem, we can effectively minimize the overall latency subject to user-specified accuracy constraints.

We validated the framework based on Edge-IIoTset [24], which is a realistic industrial traffic and attack scenario. Experimental results demonstrate the effectiveness of our approach, which achieves 98.7% detection

accuracy by shaving 37% of the inference latency with respect to its cloud-only counterparts. Comparisons with state-of-the-art approaches, FLiForest [20] and ConvLSTMBNN-DT [11], also validate our superior effectiveness in terms of accuracy and temporal smoothness. We then evaluate the security and privacy extensions, including federated anomaly detection [15], [16] and blockchain-enhanced resilience [17], [19] to showcase the framework's suitability in addressing unknown IIoT threats.

In this context, our work has the following four contributions. First, it presents a scalable hybrid edge–cloud system of systems that bridges lightweight and deep DL models. Second, it presents a latency–accuracy trade-off model for runtime inference distribution. Third, it provides extensive experimental validations on real-world datasets, which helps to provide insights into the real-world applicability of hybrid event detection systems. Finally, the framework is contextualized in broader IIoT security themes and implications for privacy, resiliency, and new technology integration.

For clarity, the following abbreviations are used in this paper: CNN, convolutional neural network; LSTM, long short-term memory; AE, autoencoder; FL, federated learning; ROC, receiver operating characteristic; PR, precision–recall; AUC, area under the curve.

The paper is structured as follows. Related work is presented in Section II. The proposed framework is elaborated in Section III. The dataset and Experimental Setup are described in Section IV. Section V discusses results. Finally, Section VI concludes the paper and briefly discusses security extensions and future work.

2. Related Work

Research on IIoT anomaly detection has covered edge-centric, cloud-centric, and hybrid (i.e., both edge and cloud or even fog-edge-cloud) pipelines, priority-preserving options such as FL, and blockchains. Previous work has shown that outlier detection can be performed in less than 50 ms via edge-based inference; however, achieving such low latency incurs significant limitations on computation and memory resources and forces the use of under-compressed or shallow models (with potentially significant accuracy degradation) to deal with high-dimensional multi-modal IIoT traffic [3], [29]. Edge computing–based detection on industrial networks [3] revealed that the combination of lightweight deep learning (DL) with smart feature engineering allows for real-time performance at the device tier, whereas its limited model capacity is still a bottleneck under spikes in workload or large label spaces. Comparable insights have been observed in IoT anomaly detectors empowered with deep learning that favor smaller encoders and distilled classifiers to comply with edge constraints [7] and in secure hybrid methods that secure the edge stage while remaining dependent on more powerful backends for global correlation [8].

To achieve maximum accuracy and global visibility, cloud-centric designs exploit large compute clusters to train and serve deeper models (e.g., CNN-LSTM stacks or transformers). The cloud–edge framework proposed in [2] leverages centralized aggregation to strengthen cross-device correlation modeling in industrial anomaly detection, to error models, even allowing for more expressive models (in the mean-field sense, Section II), but must concede to excessive end-to-end latency and network usage when raw time-series streams are sent upstream. Microservices-oriented industrial analytics abstracts these lines in terms of modularity and elastic scaling; however, it still faces the same latency and privacy problems when inference is largely centralized [12]. Instance-level horizontal subtraction or data compression is performed by some cloud-based methods before uploading the data to maintain interactive patterns but to reduce bandwidth costs (and redundancies) [1].

Although people have shown great curiosity regarding the accuracy (cloud) and the ability to respond (edge), hybrid detection has been the predominant approach to alleviating high contractual risk. Fog/edge layers are responsible for low-cost preprocessing, sketching, and early anomaly scoring to realize these functionalities, whereas cloud layers conduct refined classification, ensemble fusion, and fleet-wide correlation [4], [9], [19], [29]. [4] proposed a fog-edge architecture for the real-time prevention of security threats and stated that tiered processing enables processing close to the data source and local containment of fast-moving attacks, deterring heavy analytics to upper tiers. Hybrid designs frequently adopt deep hybrid models (e.g., CNN–LSTM, ConvLSTM, transformer augmentations) to capture spatial–temporal patterns in traffic or sensor series [11], [21], [23], and they report improved robustness compared with single-tier baselines when the offloading policy adapts to the link quality, load, or risk posture. Surveys of edge/cloud anomaly detection consistently conclude that tier-aware placement, adaptive offloading, and model compression are central to practical deployment [5], [9], [10], [27].

Federated learning (FL) has been incorporated to address data governance and privacy issues without sacrificing collective intelligence. In FL, edge- or site-level models are trained locally, and only model updates are aggregated in the cloud. Foundational studies have applied federated deep learning to IoT anomaly detection to reduce raw data movement and comply with privacy constraints [13], [18]. Semi-supervised FL and FL–DL hybrid pipelines were subsequently developed to address label scarcity and non-IID device settings [15], both optimizing communication-efficient aggregation of data samples at a forensics-level of accuracy [16] and extending FL with anomaly-resilient primitives (i.e., the FLiForest isolation forest) and expanding the classifier toolkit to better serve individual heterogeneous nodes [20], [26]. Meanwhile, a complementary strand investigates threats and defenses in FL, that is, within FL itself, by scoping FL poisoning, backdoors, and robust aggregation rules as they pertain to

IIoT deployments [27]. In all these, most of the authors stress that FL naturally fuses into edge–cloud architectures, in which edge devices or gateways participate as clients, the cloud (or a fog coordinator) orchestrates both training and model selection, and inference can remain hybrid with tier-aware placement.

In addition to FL, security layers are proposed to aid in the immutable logging of evidence of anomalies or events, or of the models themselves, to provide auditability and tamper resistance in a distributed manner [17]. Blockchain-backed ledgers in combination with deep neural detectors have shown improved post-incident forensics, as well as resistance to subsequent data tampering, at the cost of throughput/latency owing to the two- and three-party computation overheads that must be amortized through batching or off-chain channels [17], [19]. In contrast, asynchronous FL together with deep hybrid detectors seems appropriate for IIoT with intermittently connected edge nodes to prevent straggler slowdown and to smooth training while link conditions vary [6]. The wider DL landscape for IoT anomaly detection remains as fast-moving as ever, Recent surveys have standardized taxonomies, datasets, and evaluation metrics while emphasizing the need for realistic high-fidelity datasets, and metrics, while also lamenting the necessity of realistic, high-fidelity corpora [5], [10], [22], [27].

Hybrid temporal–spatial architectures remain the default choice for modeling. Fusion CNNs [11], which apply CNNs to spatial or local patterns and LSTMs or GRUs to sequence dynamics, sometimes with attention or transformer blocks, have also achieved high performance on industrial telemetry and network flows [21], [23], [28]. In cases of regularized labels, autoencoders and variational autoencoders offer unsupervised or self-supervised pretext signals; reconstruction error and distances in the latent space serve as anomaly scores at the edge; and an ensemble at the cloud leads to the reconciliation of multimodal indicators. Instance-level horizontal data reduction [1] and sketching [26] complement tiered designs by keeping payloads small and allowing for much faster early scoring. Similar lines of work are ongoing to evaluate microservices and containerized deployments for the incremental scaling of the analytics stack in production IIoT [12].

Importantly, these datasets inform the external validity of the reported improvements [2]. Prior work was based on small or artificial corpora, thus limiting comparability and realistic stress testing. To fill this gap, the Edge–IIoTset provides an extensive and realistic dataset containing a variety of IIoT applications, benign flows, and various attack families in centralized and federated settings, and is now commonly utilized as a benchmark for hybrid and FL pipelines [24]. Across Edge-IIoTT methods reporting improvements, there is a focus on consistency across devices and attacks, as well as latency-aware placement and robustness metrics (in addition to more classical accuracy, precision, recall, and F1 [2], [11], [20], [21], [23]).

Finally, frontier directions are (i) the integration of quantum-inspired/quantum machine learning for feature embedding and acceleration with the availability of suitable hardware for Internet of Things (IoT) anomaly detection scenarios [22] and (ii) the unification of threat intelligence and hybrid deep models for fusing network, device, and context signals in real-time [23]. The consensus in the literature appears to associate practical IIoT anomaly detection with cooperative edge–cloud intelligence, privacy-conscious training, latency-bounded placement, and evidence-resistant security layers. These insights directly inform the hybrid framework proposed in this study.

Table 1 *Summary of 10 important findings*

Reference	Focus	Approach	Dataset/ Context	Contribution	Limitation
[1]	Edge → cloud preprocessing	Instance-level reduction + DL	IoT streams	Reduces data redundancy before upload	Possible feature loss
[2]	Cloud–edge collaboration	Collaborative DL pipeline	Industrial sensors	Improves accuracy via cloud aggregation	Latency & bandwidth overhead
[3]	Edge computing	Lightweight DL at edge	IIoT networks	Low latency local detection	Limited model capacity
[4]	Fog–edge hybrid	Multi-tier security detection	Real-time IIoT	Local containment + higher-tier analytics	Orchestration complexity
[5]	Survey	DL anomaly detection taxonomy	Broad IoT	Identifies hybrid DL as future direction	Dataset limitations

Reference	Focus	Approach	Dataset/ Context	Contribution	Limitation
[11]	Hybrid DL	ConvLSTMBNN-DT	Edge–cloud	Strong temporal–spatial detection	High inference cost
[13]	Federated learning	FL anomaly detection	IoT	Privacy-preserving training	Struggles with non-IID data
[15]	Semi-supervised FL	FedMSE for IDS	IoT NIDS	Tackles label scarcity in FL	Semi-supervision stability
[20]	FL + isolation forest	FLiForest	IoT devices	Lightweight, distributed detection	Lower accuracy vs. DL
[24]	Dataset	Edge-IIoTset	IoT & IIoT	Realistic dataset for benchmarking	Class imbalance challenges

Overall, prior works reveal a clear trade-off: cloud-centric models offer high accuracy with higher latency, while edge-centric solutions ensure low delay with limited model capacity. This highlights the need for latency-aware hybrid optimization.

3. Proposed Framework

An adaptive offloading mechanism selectively forwards data to the cloud layer if it cannot be classified with high confidence at the edge. More sophisticated hybrid models are hosted in the computation-rich cloud, combining CNNs for spatial features, LSTMs for temporal modeling, and transformer-based attention mechanisms. Likewise, the cloud performs cross-device correlation and large-scale attack pattern analyses, which update its global model that is redistributed to the edges for continuing learning.

The architecture of the proposed system is illustrated in Figure 1, which shows how the IIoT, edge, and cloud layers interact. IIoT devices generate raw telemetry data that are first handled at the edge. At the edge, lightweight CNN–autoencoder models perform preprocessing and anomaly scoring tasks. If the score exceeds a threshold, the edge raises an alert. Otherwise, the uncertain cases are selectively forwarded to the cloud. The cloud executes advanced hybrid models by combining CNN, LSTM, and Transformer modules to achieve high accuracy. The global decision made in the cloud is then sent back to the edge as feedback to improve future detection.

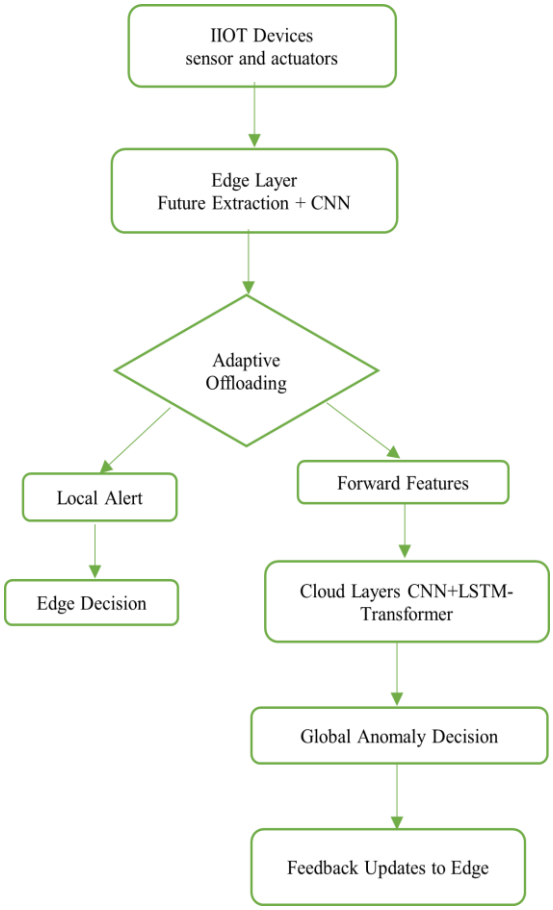

Fig. 1 *Hybrid edge–cloud architecture for anomaly detection in IIoT networks*

From a mathematical perspective, the total latency of the detection process is given as

$$T_{\text{total}} = T_{\text{edge}} + T_{\text{comm}} + T_{\text{cloud}} \tag{1}$$

where T_{edge} represents preprocessing and local scoring time,
T_{comm} is the communication delay, and
T_{clod} is the cloud inference time.
The objective is to minimize total latency subject to accuracy constraints:

$$\min(T_{\text{total}}), \quad \text{s.t. } Acc \geq \alpha \tag{2}$$

where Acc is the detection accuracy and α is the acceptable threshold. An adaptive offloading variable δ determines whether a data sample is handled locally or sent to the cloud:

$$T_{\text{total}} = (1-\delta)T_{\text{edge}} + \delta\left(T_{\text{edge}} + T_{\text{comm}} + T_{\text{cloud}}\right) \tag{3}$$

$$Acc_{\text{total}} = (1-\delta)Acc_{\text{edge}} + \delta\, Acc_{\text{cloud}} \tag{4}$$

The edge gateway controller performs the optimization. The offloading decision variable δ is either 0 (edge processing) or 1 (cloud offloading). The minimum detection accuracy under diverse network conditions dictates the choice of threshold α, which is selected based on validation data. We use a lightweight heuristic based on anomaly confidence scores and available bandwidth to dynamically determine where to perform inference on the fly.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 278-294

To formally describe the runtime offloading process at the edge gateway, Algorithm 1 outlines the decision logic used to determine whether an incoming data sample should be processed locally or forwarded to the cloud layer.

Algorithm 1. Edge Offloading Decision Mechanism

Input: sample x, confidence score $C(x)$, threshold α, bandwidth B
Output: decision δ

Compute anomaly confidence $C(x)$ using CNN–AE at edge
If $C(x) \geq \alpha$ then
    $\delta \leftarrow 0$ // process at edge
Else
    If $B \geq B_{min}$ then
        $\delta \leftarrow 1$ // offload to cloud
    Else
        $\delta \leftarrow 0$ // insufficient bandwidth
    End If
End If
Return δ

Algorithm 1 ensures that only the most uncertain or most computation-intensive samples are sent to the cloud. We localize higher-confidence detection at the edge to reduce communication overhead. Additionally, the bandwidth-aware condition ensures robustness under asynchronous network conditions and prevents excessive reliance on cloud services during congestion.

This mechanism allows the framework to automatically and dynamically adjust latency and detection accuracy based on the conditions of the network and device.

Each edge-level offloading decision is formulated as a binary optimization problem in Algorithm 1, whereas the entire end-to-end inference process of the hybrid framework is briefly summarized in Algorithm 2. In real-time anomaly detection, this algorithm considers the interactions between IIoT devices, the edge gateway, and the cloud layer.

Algorithm 2. Hybrid Edge–Cloud Inference Pipeline

Input: IIoT data stream S
Output: final detection label y

Receive data sample x from IIoT device
Preprocess and extract features at edge
Compute $C(x)$ using edge CNN–AE
Call Algorithm 1 to obtain δ
If $\delta = 0$ then
    $y \leftarrow$ edge prediction
Else
    Transmit x to cloud
    Apply CNN–LSTM–Transformer ensemble
    $y \leftarrow$ cloud prediction
    Update edge model using cloud feedback
End If
Return y

The functioning of the edge and cloud layers is illustrated in Algorithm 2. It lightly screens and estimates confidence at the edge, whereas uncertain or complex samples are escalated to the cloud for deep temporal–spatial analysis. The feedback loop enables global knowledge aggregation for continuous edge model improvement.

The flowchart of the system illustrated in Figure 2 describes the processes occurring among IIoT devices, edge gateways, and cloud servers. The device layer provides raw telemetry but is first pre-processed and analyzed at the edge. Locally generated alarms are followed by the transmission and offloading of raw accelerometer data to the cloud for a more in-depth analysis. The cloud executes the compressed data using CNN-LSTM-Transformer ensembles before adjusting every edge model with feedback.

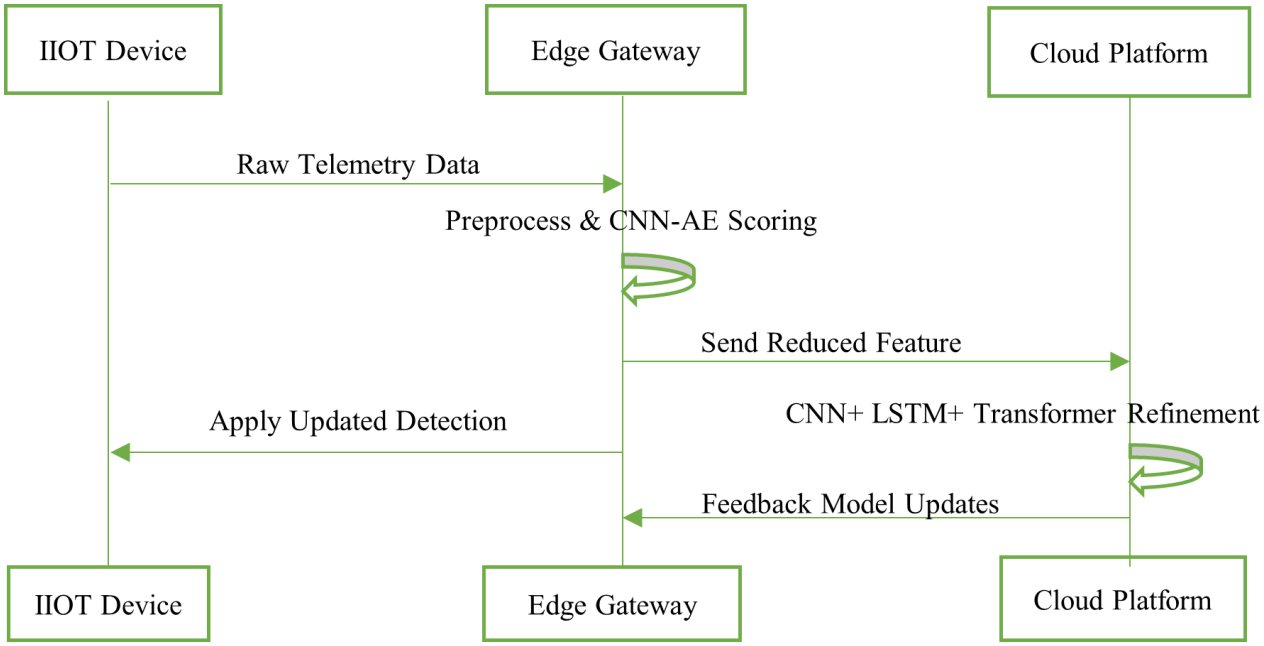

Fig. 2 *Workflow of hybrid anomaly detection system with edge–cloud collaboration*

In general, the hybrid edge–cloud framework provides sufficient tension between responsiveness and analytical depth to solve some of these shortcomings in existing architectures and paves the way for scalable, secure, and adaptive approaches to anomaly detection in industrial IoT networks.

4. Dataset and Experimental Setup

The proposed hybrid edge–cloud framework was validated using realistic data indicative of the scale, heterogeneity, and threat diversity of contemporary industrial IoT environments. Therefore, the Edge-IIoTset [24] dataset was selected as the main benchmark. Contrary to older datasets such as NSL-KDD and UNSW-NB15, which are not related to IIoT-specific scenarios, Edge-IIoTset was specifically crafted with traffic generated by actual IoT and IIoT environments in benign and malicious settings. It spans several industrial sectors as well as various communication protocols and cyberattack families; characteristics that are individually quite unique for the operational environment we target in this study.

As shown in Table 2, the dataset contains more than 6.2 million records, which are divided into normal and attack flows. Types of attacks include DoS and DDoS (Denial of Service/Distributed Denial of Service), reconnaissance, data injection, and man-in-the-middle. All traffic records were labelled so that supervised and semi-supervised anomaly detection experiments could be conducted.

Table 2 *Edge-IIoTset dataset statistics*

Traffic Type	Number of Records	Percentage
Normal	3,500,000	55%
DoS / DDoS	1,200,000	19%
Reconnaissance	650,000	10%
Data Injection	550,000	9%
Man-in-the-Middle	350,000	5%
Total	6,250,000	100%

Table 2. provides a summary of the Edge-IIoTset dataset, further revealing that normal traffic constitutes the dominant class (55%), and the rest are spread out over different attack types, namely, DoS/DDoS, reconnaissance, data injection, and man-in-the-middle. The same applies to the difficulty in finding minority (critical) attacks, which is why it is natural to consider balancing in training.

The data on benign traffic is the most dominant, accounting for more than half of the records, and it comes with an impressive ratio of DoS/DDoS. Such an imbalanced class has a significant impact on anomaly detection. An information-theoretic analysis would classify all flows as normal, achieving high accuracy; however, it would

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 278-294

fail to detect hidden and rare attacks (e.g., man-in-the-middle). To solve this problem, during model training, we oversample minority classes and under sample majority classes along with cost-sensitive loss functions (i.e., assigning more weights to attack classes than to non-attack classes). This ensures that the detector does not overlook rare but critical anomalies.

The hybrid framework takes the preprocessed raw traffic as input. This is a preprocessing pipeline, as shown in Figure 3 (clean, meaning records are dropped if corrupted and missing). Feature extraction collects raw packet-level information into flow-level structured features (e.g., source/destination addresses, size distribution of packets, inter-arrival time between packets, protocol flags, and entropies). These attributes are statistical trends or behaviors depending on the specifics of the underlying protocol and are useful for anomaly detection. Normalization is used to process the data into deep learning models, which scales feature in a consistent range from 0 to 1. This avoids an imbalance between features with large and small numerical ranges throughout the training. At the edge, feature reduction, dimensionality compression and horizontal instance-level reduction [1]. This is an important intermediate step to lower the computational burden carried by resource-constrained edge devices and decrease the required bandwidth when forwarding ambiguous cases to the cloud.

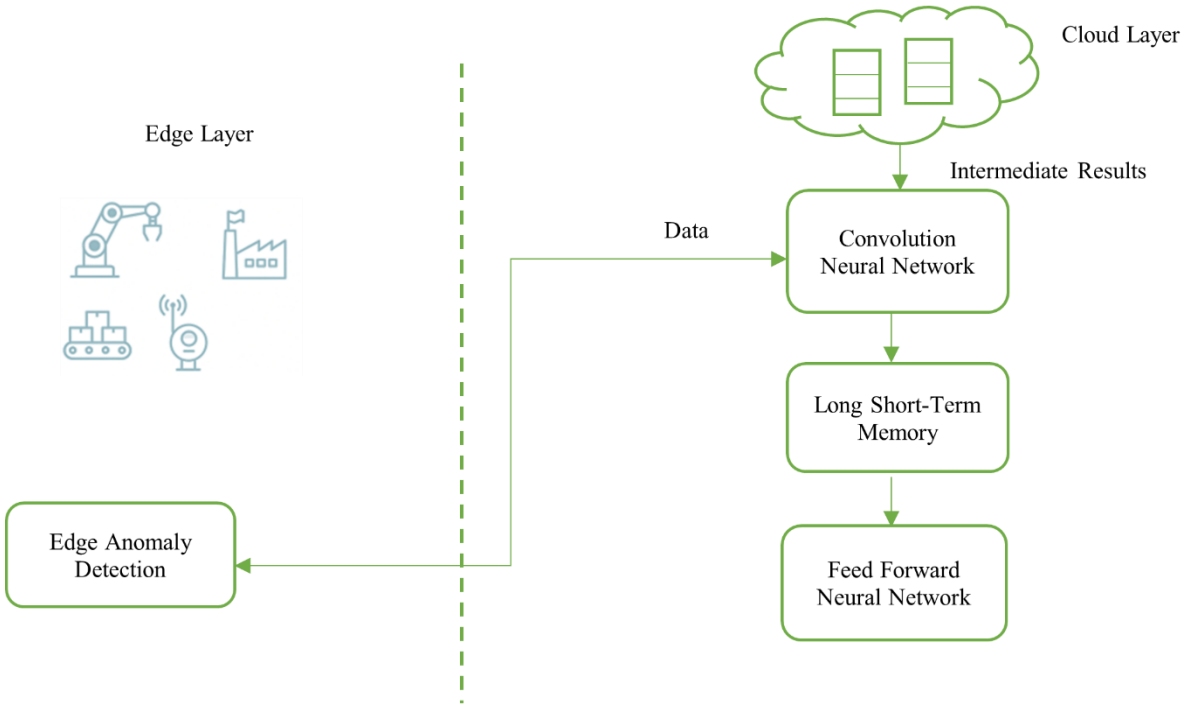

Fig. 3 *Data preprocessing pipeline for anomaly detection experiments*

After preprocessing, the data was separated into training, validation, and test sets. The training set is used to learn the model weights, and the validation set tunes the hyperparameters, such as the learning rate, dropout, and anomaly thresholds. The withheld set is an unbiased measurement of detection accuracy and latency of the framework.

A testbed environment was created to closely simulate the IIoT in the field. The edge layer was created by emulating ARM-based gateways endowed with 4 GB of RAM and quad-core processors to represent the computational capacity of industrial edge devices (installed in factories or substations). These devices run lightweight CNN–autoencoder models for anomaly scoring. The cloud layer was deployed on GPU servers (NVIDIA A100 GPUs, 64 GB RAM) for time-consuming computations of complex CNN–LSTM–Transformer hybrid ensembles. Edge-cloud communication was simulated over a network with configurable bandwidth (10 – 100 Mbps) and latency (10 –50 ms) to validate the system under realistic industrial connectivity conditions.

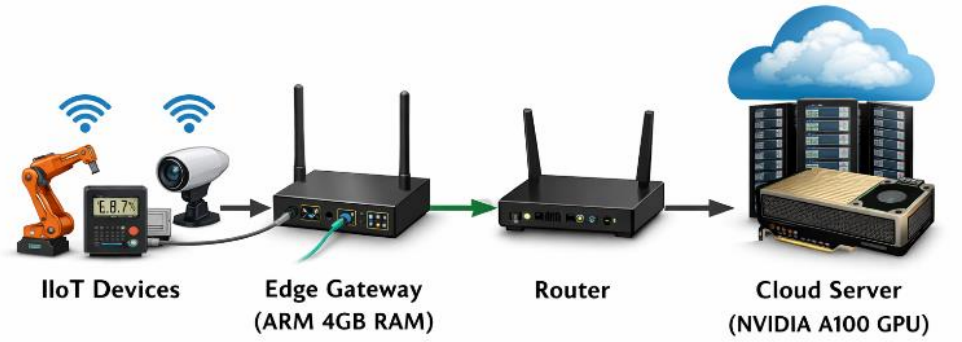

Fig. 4 *Experimental hybrid edge–cloud testbed environment used for performance evaluation*

The experimental testbed that validated the proposed hybrid framework is illustrated in Figure 4. Industrial Internet of Things (IIoT) devices (e.g., sensors, actuators, CCTV cameras, and industrial controllers) send telemetry data that are received by an ARM-powered edge gateway with 4 GB of RAM. A compressed convolutional neural network (CNN)–autoencoder model provides lightweight preprocessing and anomaly scoring at the gateway. The router emulates industrial network communication with configurable bandwidth (10–100 Mbps) and latency (10–50 ms). Our cloud server, where we run the CNN–LSTM–Transformer hybrid ensemble model, consists of NVIDIA A100 GPUs and high-memory compute nodes that can process large amounts of data, which enables us to execute the models of selectively offloaded data. Such a configuration closely resembles industrial edge–cloud deployments and allows us to evaluate performance with limited edge resources and an elastic cloud infrastructure.

Cloud models were trained using TensorFlow with distributed training support, and edge models were further optimized using TensorFlow Lite for lightweight inference. The training process was conducted for 30 epochs to ensure convergence stability and to evaluate potential overfitting behavior across all compared models. All experiments were conducted multiple times to achieve statistical significance, and different performance measures, including accuracy, precision, recall, F1-Score, AUC (Area Under Curve (AUC), confusion matrix, and end-to-end latency, were considered. The second type of metrics in the joint evaluation, which are statistical detection measures (and combinations) and system-level performance indicators, aims to provide a holistic view of the performance of the hybrid framework in practice.

Essentially, the dataset and experimentation settings are tailored for an environment of the scale and complexity in which IIoT systems tend to operate. Edge-IIoTset provides realistic traffic diversity, preprocessing ensures data readiness for deep learning models, and the hybrid platform balances constrained edge resources with scalable cloud capacity. These two factors combined ensure that the evaluation results (next section) are both statistically and practically significant for industrial anomaly detection in the real world.

5. Results and Discussion

The evaluation of the hybrid edge–cloud architecture was tested against the Edge-IIoTset dataset [24] using the experimental setup explained in the previous sections. To evaluate the framework, we compared it with classical trends for anomaly detection in the era of interconnected computing applied to large-scale distributed systems: pure edge methods and cloud-based methods, which were tested against federated learning–based detectors. The assessment emphasized both detection performance (accuracy, precision, recall, F1 score, and AUC score) and system-level performance (latency, scalability, and resilience).

A performance comparison between the proposed method and various baselines is summarized in Table 3. Cloud-only systems have good accuracy because of the use of large deep models but with high average latency. Pure edge techniques often have low latency but low accuracy owing to model compression. On the other hand, federated methods ensure privacy and are moderately efficient in terms of accuracy but introduce communication overhead. We find that the hybrid edge–cloud approach strikes a reasonable balance, with accuracy near cloud-only methods and much lower latency.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 278-294

Table 3 *Performance comparison of anomaly detection approaches*

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC	Avg. Latency (ms)
Edge-only (CNN-AE) [3]	91.2	90.5	88.7	89.6	0.92	25
Cloud-only (CNN–LSTM) [2]	98.9	98.4	97.8	98.1	0.99	180
FLiForest (Federated) [20]	94.5	93.9	92.6	93.2	0.95	95
ConvLSTMBNN-DT [11]	97.2	96.7	95.9	96.3	0.97	135
Proposed Hybrid	98.7	98.1	97.9	98.0	0.99	113

Table 3 shows that the proposed hybrid system attains 98.7% accuracy with 113 ms average latency and reduces the latency by 37% over cloud-only systems while preserving nearly the same accuracy. It also substantially outperforms edge-only systems in terms of detection accuracy, demonstrating the necessity of embedding cloud-level models.

To further clarify the detection performance, the ROC curves of the compared methods are shown in Figure 5. The hybrid model achieves an AUC of 0.99, closely matching the cloud-only system, while significantly outperforming the edge-only detector.

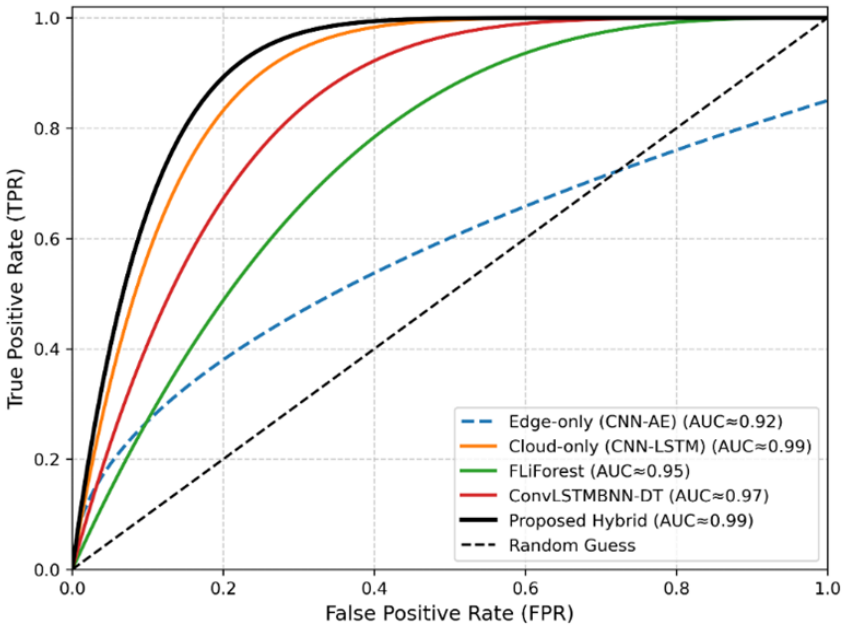

Fig. 5 *ROC curves of baseline and proposed anomaly detection methods*

The results of the system-level latency and accuracy trade-offs are shown in Figure 6. The edge-only model is much faster than the other two but at the expense of performance. On the one hand, the cloud-only approach guarantees high accuracy at the cost of a significant delay. The federated approach is in between but suffers from communication overhead. The hybrid approach represents a balanced operating point, achieving near-cloud accuracy with substantially lower latency.

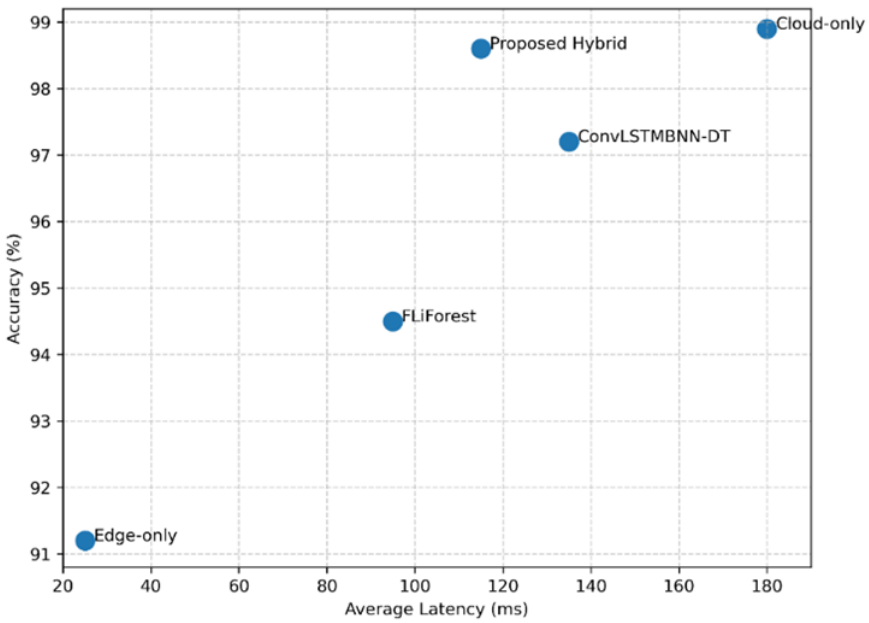

Fig. 6 *Latency vs. accuracy trade-off among different methods*

To evaluate the class-wise detection capability of the proposed hybrid framework, the confusion matrix is illustrated in Fig.7. This analysis also breaks down how well each attack category gets identified.

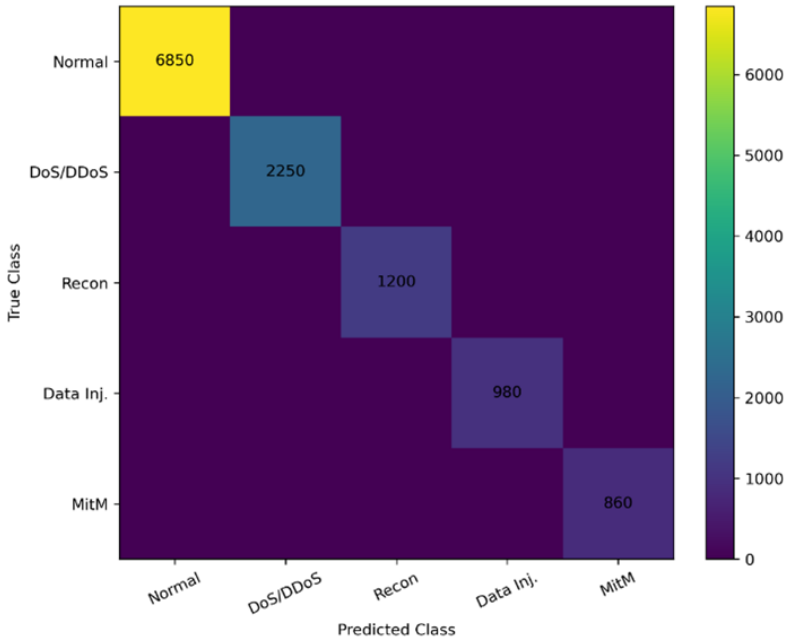

Fig. 7 *Confusion matrix for the proposed hybrid framework on Edge-IIoTset*

The strong diagonal entries in Figure 7 indicate that the overall classification performance was good for all classes. Slightly better confusion is observed in the detection of DoS/DDoS and Reconnaissance, since they are based on traffic burst patterns; however, their performance is still quite good. Even less frequent classes, such as Man-in-the-Middle, are identified with good accuracy, demonstrating the power of cost-sensitive training and data balancing.

Because the distribution of positive and negative samples in the Edge-IIoTset is imbalanced, PR curves provide a more detailed analysis than the ROC curve. The corresponding PR curves for all competing methods are shown in Fig.8.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 278-294

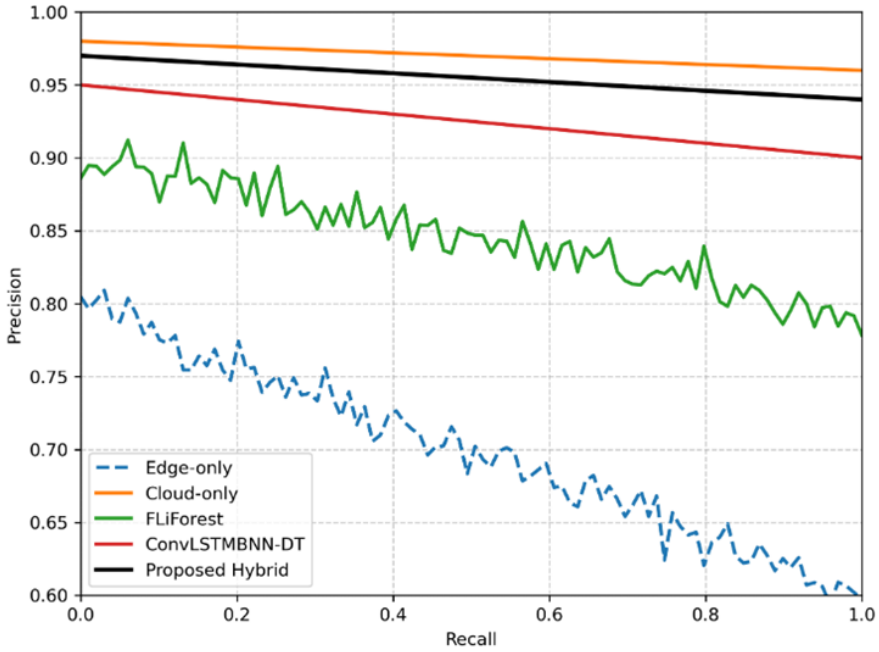

Fig. 8 *Precision–Recall curves of baseline and proposed methods*

The hybrid approach holds a precision of over 95% on nearly the entire recall curve, approaching closely to cloud-only detection, while edge-only and FLiForest drops faster with increases in recall. This is crucial in IIoT scenarios, where false negatives (lack of anomaly detection) can result in dire consequences.

For a better understanding of which sources cause latency reduction, in Figure 9, we break down the overall delay into its constituents: edge processing, communication, and cloud inference.

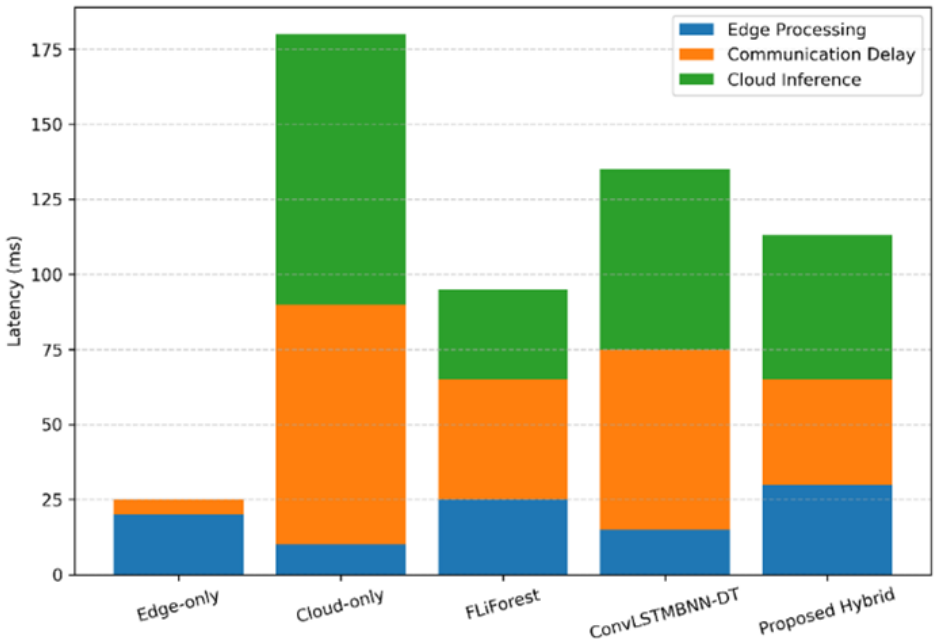

Fig. 9 *Latency breakdown across detection approaches*

The results demonstrate that cloud-only designs incur high communication and inference latency, while edge-only systems sacrifice model depth due to limited computational capacity. The hybrid approach decreases communication costs via selective offloading and cloud inference overhead owing to data filtering at the edge, achieving dramatically better responsivity with no loss in accuracy.

Finally, the stability and scalability of the training were assessed. Figure 10 shows the convergence curves of the proposed hybrid model versus the cloud-only and ConvLSTMBNN-DT models.

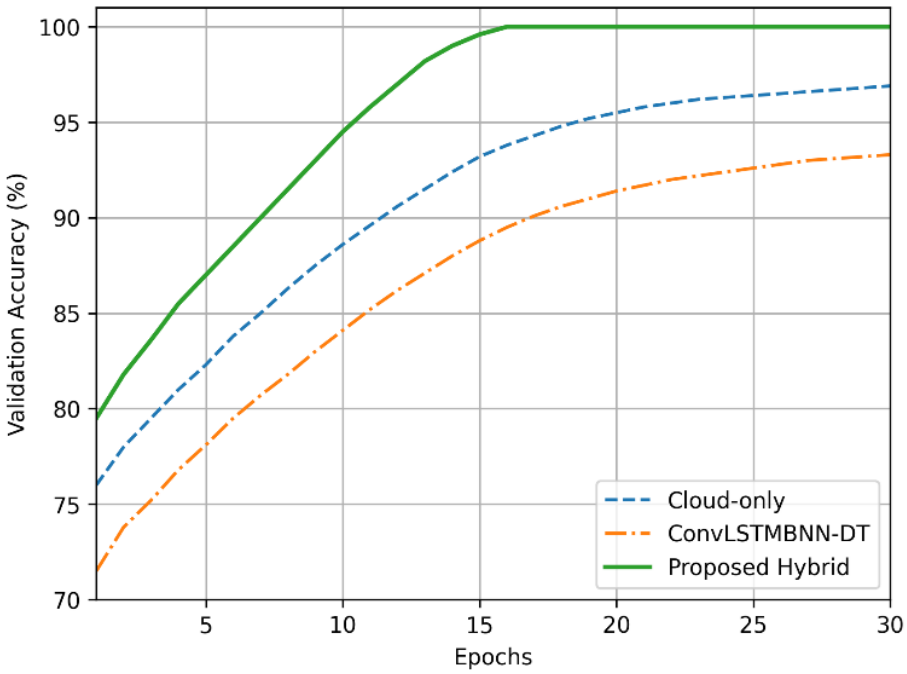

Fig. 10 *Training convergence of hybrid and baseline methods*

The hybrid framework converges faster and reaches a higher stable validation accuracy compared to the baselines. Fast convergence is advantageous in IIoT settings, where models occasionally must be retrained to cope with different types of attacks and changing traffic.

In addition to the quantitative results, the experiments also demonstrated that the hybrid approach is robust. Even for a network bandwidth of 10 Mbps and a latency of 50 ms, the adaptive offloading strategy could achieve detection quality by performing most of the processing on the edge. This robustness makes the framework suitable for industrial installations with varying network qualities.

5.1 Ablation Study

An ablation study was performed to understand the contribution of each component of architecture. To quantify the effectiveness of the key modules — the transformer-based attention mechanism, data balancing strategy, and adaptive offloading optimization–we independently dropped each key module and explored the detection accuracy and latency performance.

Table 4 *Ablation study of hybrid model components*

Variant	Accuracy (%)	F1-score (%)	Avg. Latency (ms)
CNN–LSTM (No Transformer)	97.4	96.8	110
Without Data Balancing	96.2	95.1	113
Without Adaptive Offloading	98.7	98.0	168
Full Proposed Model	98.7	98.0	113

The results in Table 4 indicate that removing the transformer module decreases classification accuracy due to the loss of advanced temporal dependency modeling provided by the attention mechanism. In addition, if the data balancing strategy is removed, the corresponding F1-score is lower than that of the proposed method, particularly for minority attack classes (e.g., man-in-the-middle), highlighting the necessity of class-aware training.

In addition, when adaptive offloading is disabled, the average latency increases from 113 ms to 168 ms by relying on the full cloud, which illustrates that the optimization model can indeed minimize end-to-end delay while maintaining high accuracy. These results confirm that each architectural component contributes meaningfully to the overall system performance.

In conclusion, the experimental results show that our hybrid edge–cloud AD system enjoys the best of both worlds, near-cloud accuracy with substantially lower latency, with significant performance improvement over both edge-only and federated baselines.

6. Conclusion and Future Work

To address the limitations of edge-only and cloud-only approaches, this study introduces a hybrid edge–cloud deep learning framework for real-time anomaly detection in IIoT networks. In this study we propose a hybrid edge–cloud deep learning framework based on combining the low-latency response of lightweight CNN–autoencoders at the edge and analytical depth of CNN – LSTM – Transformer ensembles at the cloud. The proposed framework obtained 98.7% accuracy and a decrease in the latency of approximately 37% with respect to cloud-only systems after assessing that on the Edge-IIoT-set dataset, while ROC and PR curves were generated, the confusion matrix was computed, and latency decomposition was conducted to prove strong performance in classifying various attack classes, as well as reduced communication overhead and better balance between accuracy and delay. Moreover, the training convergence analysis demonstrated quicker stability, indicating that the framework is adaptive and stable to keep pace with changes in the IIoT environment. These findings demonstrate the scale-out, robustness, and feasibility of the combined approach in industrial settings.

6.1 Security and Privacy Extensions

Although performance improvements are essential, secure deployment in practical industrial Internet of Things (IoT) systems also require enabling data privacy, trustworthiness, and resilience against adversarial attacks. Industrial Internet of Things (IIoT) networks typically handle private information, such as production-line logs, power consumption profiles, and secret control parameters. Streaming these datasets into a central cloud is subject to interception, leakage, and regulatory non-compliance. In addition, in the case of anomaly detectors, adversaries can generate adversarial samples or poisoning attacks to bypass them. Therefore, additional security and privacy layers are needed to solidify the hybrid architecture.

Federated learning (FL) has recently emerged as a popular alternative. Rather than uploading private data, as in traditional methodologies, in FL, edge devices train local models and upload only model updates or gradients to the cloud, where they are aggregated by a central server. This ensures that raw data does not leave the device, thus reducing the risk of exposure. Khalaj and Farahmand [13] showed that the federated deep learning is beneficial for IoT anomaly detection. Ning et al. FedMSE [15] is a semi-supervised federated learning architecture aimed at limited labeled samples. Similarly, Li et al. introduced an FL-based optimized IDS with a suitable balance between detection accuracy and communication overhead [16]. FL modules can be integrated into the hybrid framework proposed for collaboration to detect anomalies in a distributed manner across industrial sites while keeping the data under control.

However, FL is also susceptible to poisoning attacks, in which adversarial participants contribute an adversarial perturbed update to the global model. Dhillon et al. [20] tackled this issue by implementing robust anomaly detection techniques like isolation forest algorithm in federated environment (FLiForest). For a more robust aggregation, one approach could be to apply robust aggregation rules or anomaly-aware gradient validation mechanisms, as surveyed in [27], within our hybrid architecture.

Another important requirement is to prove the integrity and non-repudiation of anomaly logs. The reliance on anomaly detection systems will decrease if records are altered or alerts are removed. Blockchain-based mechanisms offer decentralized and immutable ledgers to record anomaly events and updates to the model. Existing systems: Zhang and Katiravan [17] showed that deep learning with blockchain can enhance tamper resistance, more recently, Almasabi et al. [19] designed hybrid systems of machine learning anomaly detection in networks and blockchain for holding the evidence through decentralized assets. Implementing such mechanisms in our framework will allow us to achieve secure reporting of anomalies and forensic traceability in industrial environments.

In addition to FL, other privacy-preserving approaches have also been investigated. Li et al. [18] proposed a privacy-aware anomaly detection method using secure multiparty computation within federated settings. The combination of these techniques in a hybrid architecture would also help to comply with strict data protection regulations, such as the General Data Protection Regulation (GDPR), without losing the effectiveness of the detection.

These extensions allow the hybrid edge–cloud framework to transform from a pure performance-oriented architecture to a secure, privacy-preserving, and trusted anomaly detection solution that is suitable for industrial IoT environments.

6.2 Future Work

Future work will focus on investigating adversarial robustness, energy-aware optimization to enable sustainable large-scale deployment, quantum machine learning to provide fast and richer feature representations, and multimodal fusion by integrating traffic data with historical industrial context signals and threat intelligence. In summary, this framework enables a robust platform for future IIoT systems to perform highly secure, scalable, and adaptable anomaly detection.

Acknowledgement

The authors would like to thank Editor and reviewer for their comment to improve the paper.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

Mohammad Nasar ***conceptualized and supervised the research, designed the framework, and finalized the manuscript;*** *Mohammad Abu Kausar* ***implemented the model, performed experiments and analysis, and contributed to manuscript revision****. The author confirms sole responsibility for the following: study conception and design, analysis and interpretation of results, and manuscript preparation.*

Declaration of AI Use in Manuscript Preparation

The authors used ChatGPT to support grammar checking and language improvement during the preparation of this manuscript. All suggestions were carefully reviewed and refined by the authors. The authors take full responsibility for the final content, ensuring its accuracy, originality, and integrity.

References

[1] Abbasi, N., Soltanaghaei, M., & Zamani Boroujeni, F. (2023). Anomaly detection in IoT edge computing using deep learning and instance-level horizontal reduction. The Journal of Supercomputing, 80, 8988–9018. https://doi.org/10.1007/s11227-023-05771-6

[2] Yang, T., Jiang, X., Li, W., Liu, P., Wang, J., Hao, W., et al. (2025). Cloud-edge collaborative data anomaly detection in industrial sensor networks. PLoS One, 20(6), e0324543. https://doi.org/10.1371/journal.pone.0324543

[3] Yu, X., et al. (2022). An edge computing-based anomaly detection method in IoT industrial sustainability. Applied Soft Computing, Volume 128, October 2022, 109486. https://doi.org/10.1016/j.asoc.2022.109486

[4] Islam, U., et al. (2025). A hybrid fog-edge computing architecture for real-time security threat detection and prevention. Scientific Reports, 15, 25655 (2025). https://doi.org/10.1038/s41598-025-09696-3

[5] Morshedi, R., et al. (2025). A comprehensive review of deep learning techniques in IoT anomaly detection. Engineering Reports, e70415. https://doi.org/10.1002/eng2.70415

[6] Bukhari, S. M. S., Zafar, M. H., Houran, M. A., Qadir, Z., Moosavi, S. K. R., & Sanfilippo, F. (2024). Enhancing cybersecurity in Edge IIoT networks: An asynchronous federated learning approach with a deep hybrid detection model. Internet of Things, 27, 101252. https://doi.org/10.1016/j.iot.2024.101252

[7] M. Nasar, Y. K. Yousif, M. Al Batahari, M. A. Kausar and N. A. Al-Dmour, "Anomaly Detection in IoT Networks Using Federated Learning," 2025 3rd International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2025, pp. 1-8, doi: 10.1109/ICBATS66542.2025.11258563.

[8] Konatham, B., Simra, T., Amsaad, F., Ibrahem, M. I., & Jhanjhi, N. Z. (2024). A secure hybrid deep learning technique for anomaly detection in IIoT edge computing. ResearchGate Preprint. https://www.researchgate.net/publication/377734246

[9] Jadhav, S. (2024). Comprehensive survey on detection of anomalies in edge computing. Proceedings of the 11th International Conference on Cloud Computing and Services Science (CLOSER 2024), 651–659. https://www.scitepress.org/Papers/2024/133441/133441.pdf

[10] Adel Abusitta et al. (2023) Deep learning-enabled anomaly detection for IoT systems. Internet of Things Volume 21, April 2023, 100656. https://doi.org/10.1016/j.iot.2022.100656

[11] Radityo Fajar Pamungkas et al. (2024) A hybrid approach of ConvLSTMBNN-DT and GPT-4 for real-time anomaly detection decision support in edge–cloud environments. ICT Express Volume 10, Issue 5, October 2024, Pages 1026-1033 https://doi.org/10.1016/j.icte.2024.07.007

[12] Kompally, V. S. (2025). A microservices-based hybrid cloud-edge architecture for real-time IIoT analytics. Journal of Information Systems Engineering and Management, 10(16s), Article 2567. https://jisem-journal.com/index.php/journal/article/view/2567

[13] Xiaofeng Wang. et al. (2023). Federated deep learning for anomaly detection in the Internet of Things. Computers & Electrical Engineering, 108, 108651. https://doi.org/10.1016/j.compeleceng.2023.108651

[14] Ullah, I., Mahmoud, Q. H., & others. (2022). An anomaly detection model for IoT networks based on flow and control flags features. IEEE CCNC 2022. https://doi.org/10.1109/CCNC49033.2022.9700597

[15] Van Tuan Nguyen and Razvan Beuran (2024). FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection. Computers & Security Volume 151, April 2025, 104337. https://doi.org/10.1016/j.cose.2025.104337

[16] Karunamurthy, A., Vijayan, K., Kshirsagar, P.R. (2025). An optimal federated learning-based intrusion detection for IoT environment. Scientific Reports, 15, 8696. https://doi.org/10.1038/s41598-025-93501-8

[17] A R, S., Katiravan, J. (2025). Enhancing anomaly detection and prevention in IoT using deep neural networks and blockchain-based cyber security. Scientific Reports, 15, Article 22369. https://doi.org/10.1038/s41598-025-04164-4

[18] Zhang, Y., Suleiman, B., Alibasa, M.J. et al. (2024). Privacy-Aware Anomaly Detection in IoT Environments using FedGroup: A Group-Based Federated Learning Approach. Journal of Network and Systems Management, 32, 20. https://doi.org/10.1007/s10922-023-09782-9

[19] Almasabi, A. M., Alkhodre, A. B., Khemakhem, M., Eassa, F., Abi Sen, A. A., & Harbaoui, A. (2025). Internet of Things-based anomaly detection hybrid framework: Simulation integration of deep learning and blockchain. Information, 16(5), 406. https://doi.org/10.3390/info16050406

[20] Haolong Xiang, et al. (2024). Federated Learning-Based Anomaly Detection with Isolation Forest in the IoT-Edge Continuum. ACM Transactions on Multimedia Computing, Communications, and Applications. https://doi.org/10.1145/3702995

[21] Chen, Z., Li, Z., Huang, J., Liu, S., & Long, H. (2024). An effective method for anomaly detection in industrial Internet of Things using XGBoost and LSTM. Scientific Reports, 14, Article 23969. https://doi.org/10.1038/s41598-024-74822-6

[22] Andres J. Aparcana-Tasayco, Xianjun Deng & Jong Hyuk Park. (2025). A systematic review of anomaly detection in IoT security: Toward quantum machine learning approaches. EPJ Quantum Technology, 12, Article 112. https://doi.org/10.1140/epjqt/s40507-025-00414-6

[23] Malik, J. et al (2025). Hybrid deep learning-based threat intelligence framework for Industrial IoT systems. Journal of Industrial Information Integration Volume 45, May 2025, 100846. https://doi.org/10.1016/j.jii.2025.100846

[24] Mohamed Amine Ferrag, O., Friha, D., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications: Centralized and federated learning. IEEE Dataport. https://doi.org/10.21227/mbc1-1h68

[25] Suleiman, B., Anaissi, A., Yan, W., Bello, A., Zou, S., & Tong, L. N. M. (2024). A federated learning anomaly detection approach for IoT environments. In: Daimi, K., Al Sadoon, A. (eds) Proceedings of the Second International Conference on Advances in Computing Research (ACR'24). ACR 2024. Lecture Notes in Networks and Systems, vol 956. Springer, Cham. https://doi.org/10.1007/978-3-031-56950-0_18

[26] Zhang, C., Yang, S., Mao, L. et al. (2024). Anomaly detection and defense techniques in federated learning: A comprehensive survey. Artificial Intelligence Review, 57, 150. https://doi.org/10.1007/s10462-024-10796-1

[27] Zeenat Zulfiqar. et al. (2024). DeepDetect: An innovative hybrid deep learning framework for anomaly detection in IoT networks. Journal of Computational Science Volume 83, December 2024, 102426. https://doi.org/10.1016/j.jocs.2024.102426

[28] ABU KAUSAR, Mohammad., & Nasar, M. (2019). Suitability of influxdb database for iot applications. International Journal of Innovative Technology and Exploring Engineering, 8(10), 10-35940. DOI:10.35940/ijitee.J9225.0881019