RESEARCH PAPER

FlowGuard: A Privacy-Preserving, Flow-Aware Deep Learning Framework for Real-Time Intrusion Detection in IIoT Networks

Guduru Jyothsna Devi[1], Radhika Rani Chintala[2]

[1]M. Tech CSE, student, Dept. of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522302, India. Email: 2401050073cse@gmail.com

[2]Associate Professor, Dept. of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522302, India. Email: radhikarani_cse@kluniversity.in

ABSTRACT

The environments of industrial Internet of Things (IIoT) face more advanced cyber threats, necessitating an intrusion detection system (IDS) with high accuracy, privacy protection, and the ability to withstand information leakage. To overcome these issues, this paper presents an Enhanced Intrusion Detection Framework (EIDF) which integrates flow-based data isolation, deterministic hash-based feature sanitization, SMOTE-based class rebalancing, and regularized deep sequential model. The framework is trained with the WUSTL-IIoT-2021 dataset (1,048,576 flows) containing 58 engineered features with no flow-isolated train tests, and a 80:20 flow-isolated train-test split to avoid any overlap in time or topology. Three sequential architectures are trained on 50 epochs with the Adam optimizer (lr = 0.001, batch = 32) with class-weighted binary cross-entropy, including Regularized LSTM (128-64 units, dropout = 0.3, L2 = 10^{-4}), TCN using soft attention and Hybrid CNN-LSTM. It is experimentally shown that the LSTM_Regularized model achieves 99.9809% accuracy, 0.999998 AUC, 99.8688% F1-score, and 99.7379% attack recall and can maintain real-time inference time (1.51 ms/sample) on NVIDIA Tesla P100 hardware. The independent aspects of flow-aware partitioning, deterministic hashing, and regularization are found to play a crucial role in generalization and privacy by ablation studies. The proposed EIDF sets a new standard of IIoT intrusion detection in the real sense striking a balance between detection accuracy, and practical scalability, and robustness.

Keywords: Data leakage prevention, Flow-based Splitting, IIoT Security, Intrusion Detection, Real-time security, Regularization, Sequential Deep Learning, SMOTE Rebalancing

How to cite this article: Devi GJ, Chintala RR. FlowGuard: A Privacy-Preserving, Flow-Aware Deep Learning Framework for Real-Time Intrusion Detection in IIoT Networks. Int J Drug Deliv Technol. 2026;16(49s): 973-984. DOI: 10.25258/ijddt.16.49s.113

Source of support: Nil.

Conflict of interest: None

I. INTRODUCTION:

The recent blistering development of Industrial Internet of Things (IIoT) systems has revolutionized key areas of infrastructure, including intelligent manufacturing and power grids, healthcare and transport services, with real-time automation and intelligent control. But there exist cybersecurity threats of a kind never. In contrast to traditional IT networks, IIoT systems must operate under the strictest of conditions: there are scarce computational resources, communication is deterministic, and downtime is unacceptable [4], [11]. In turn, IIoT intrusion detection requires high detection accuracy, in addition to temporal awareness, low-latency inference, and resistance to changing, stealthy attackers, such as reconnaissance, protocol abuse, and denial-of-service [15], [18].

Conventional signature or rule-matching-based Intrusion Detection Systems (IDS) are not effective in detecting zero-day attacks, and have poor false-positive rates, which makes them ineffective in dynamic IIoT systems [13]. The area of research has thus transitioned into machine and deep learning, using CNNs and LSTMs and ensemble models on datasets like AWID, NSL-KDD, and WUSTL-IIoT [1], [2], [6], [7], [8], [15]. Although the metrics are strong, there remains a methodological weakness which is the ubiquitous random or stratified train-test split that does not consider the underlying flow structure of network traffic [3], [12], [19]. This leakage creates a topological and temporal data that enables the packets belonging to the same stream of communication to be in both training and test sets. This leads to false high accuracy since models are memorized with patterns of packets and these are not learned to generalize attack behavior [15], [18].

No less problematic is the handling of network identifiers, i.e. source and destination IPs, ports, and protocols. Keeping them in their raw categorical form will result in memorization of device-specific relationships [1], [3], and removing them will remove useful contextual information [6], [7]. In the meantime, numerous intrusion datasets are grossly imbalanced (attacks 1% of traffic) and popular oversampling methods such as SMOTE are used

without keeping the time sequence, producing artificial sequences that can potentially break realistic flow semantics [12], [15], [16].

The other significant gap is a gap in evaluation rigor. Most of the studies present only aggregate statistics like accuracy or F1-score, disregarding attack-class recall (true positive rate of intrusion), which is especially important in security terms [5], [13], [18]. Very few are gauged on inference latency, although IIoT is very strict on real-time expectations [19]. Also, hybrid or attention-augmented architectures have been proposed [12], [17], although they tend not to be systematically regularised, flow-insensitive or flow-aware validation, and interpretable feature engineering, which is restrictive to robustness and deployability [4], [9], [14].

With the aim of correcting these deficiencies, this paper re-views IIoT intrusion detection as a flow-centric, leakage-free, and behaviorally based learning problem. We consider each network flow which is source-destination defined, and protocol defined as the basic unit of communication, and we strictly separate training and testing flows. Deterministic hashing is used to preserve network identifiers so that the structural relationships are retained without memorization [1], [3]. Sequential deep architectures are used that best model intra-flow temporal dynamics and are enhanced with L2 regularization, dropout, attention and class-weighted optimization to be robust to imbalance [12], [15], [18].

Lastly, we do not just evaluate using aggregate measures. We present per-class recall (especially of attacks), AUC-ROC, F1-score, and inference latency, guaranteeing security effectiveness and system feasibility [19]. Our framework offers an answer of conceptual soundness, feature sanitization, model-based dual-path sequence, and intensive benchmarking to IIoT intrusion detection.

II. RELATED WORK:

Machine learning (ML) and deep Learning (DL) have become important in intrusion detection in wireless and Industrial Internet of Things (IIoT) networks. Initial methods were based on signature/rule-based systems which were characterized by high rates of false-positive and low adaptability to new attacks. Recent studies have diverted to the use of data-driven models to enhance the accuracy and strength.

One of the major trends is the deep learning of wireless sensor and Internet of Things networks. The authors Campos et al. [1] suggested CNN, DNN, LSTM models on the AWID dataset and reached up to 98% accuracy in binary classification after feature selection. In like manner, Vyas et al. [2] applied ResNet50 with recursive feature removal to AWID to identify multi-classes and achieved an F1-score of 99.73% but overfitting was also an issue. Mesadieu et al. [3] used instance selection (SamSelect) with stacked contractive autoencoders and a Conditional Deep Belief Network to achieve an accuracy of 97.4%--but without explicit feature selection, which could result in redundancy in the high-dimensional WSN data.

A different body of research is based on hybrid and ensemble strategies. Ozkan-Okay et al. [13] proposed SABADT, a combination of signature- and anomaly-based detection using KDD and UNSW-NB15 data, with 99.65% precision using 17 selected features, but without hyperparameter optimization. Mohy-Eddine et al. [12] used an ensemble model Isolation Forest and Pearson correlation to feature engineer IIoT, proving to be highly effective on benchmark datasets--however, it is not proven to be applicable in dynamic industrial contexts.

The critical issues with IIoT are class imbalance and real-time constraints. Jiang et al. [16] used LightGBM with backward feature selection on LEACH-simulated WSN traffic achieving 99.73 percent accuracy, but tree-based models might not be applicable to high-speed and real-time packet inspection. To counter imbalance, Popoola et al. [15] suggested Multi-Stage Deep Learning (MSDL) framework on imbalanced IIoT data (X-IIoTID, WUSTL-IIoT), which outperforms the standard oversampling-but its performance against zero-day attacks is ambiguous.

There has also been investigation on architectural innovation. Yang et al. [6] and Yang et al. [7] implemented CNNs on SCADA intrusion detection, which does not require any manual feature engineering but may fail to detect non-spatial attack patterns. Seo and Pak [19], by contrast, developed a two-stage hybrid ML system: a temporal Level-1 classifier to make real time decisions and a flow-based Level-2 analyzer to classify uncertain cases--it is, however, vulnerable to single-packet attacks. Shahid et al. [18] tested ML/DL models on RPL-based IoT networks, with Random Forest (99%

accuracy) and Transformers (97% F1) performing well, though only in comparison with four known routing attacks.

New paradigms are side-channel analysis and federated learning. Campos et al. [1] tracked the power consumption of the devices to detect anomalies- the power consumption of devices is low but the equipment is expensive to deploy in large scale networks. Vyas et al. [2] reviewed the privacy-preserving federated learning (PPFL) in the context of IoT IDS and found homomorphic encryption and differential privacy, but communication overhead and convergence limitations on resource-limited devices are barriers to practice. In the meantime, Ankalaki et al. [14] investigated generative AI in predicting cyberattacks, warning that it can be used dual-purpose in creating advanced threats.

The SCADA-specific solutions tend to favor architectural novelty, as opposed to robustness. Falco et al., Shitharth et al. [4] and [17] suggested CNN-based and RPCO-BCNN-based detectors respectively with high accuracy but not as much evaluation against physical or insider threat. Hasan and Mouftah [11] optimized the placement of IDS through linear programming, modeling network topologies as static- which is not realistic in a dynamic smart grid.

Taken together, these works show the advancement but with obvious gaps: overfitting because of poor feature selection, poor generalization, and poor class imbalance and real-time processing management. We deal with them by incorporating flow awareness data partitioning, strict feature sanitizing, hybrid sequential modeling, and regularized training on a weighted loss of classes that guarantees robust, leakage free, and high-performance intrusion detection in IIoT settings.

Contributions and Novelty:

Although recent papers have conducted research on deep learning in detecting intrusion in IIoT and wireless networks, serious methodological gaps can be identified. Most methods use standard random splitting, which unintentionally leads to leakage of data at the flow level between training and evaluation [2,3,6,12]. Others do not practice identifier sanitization, putting them at risk of overfitting to fixed IP addresses or ports [1,15]. Despite some occasional use of feature selection and balancing classes [13,18], temporal sequence modelling is often used with respect to flow semantics. Also, model testing does not typically include metrics of latency that are necessary to deploy IIoT in the real-time context [4,7,19].

To overcome these shortcomings, this paper proposes the Enhanced IIoT Network Traffic Classification Pipeline, which is the first such offering integrating:

1. Group Shuffle Split with flow aware data partition to remove cross flow leakage.
2. Protecting structural patterns without memorization Feature sanitization through identifier hashing.
3. Representation learning Flow-based and token-based Dual-path sequential representation learning A rich temporal model.
4. L2 regularization, dropout and class-weighted training Hybrid deep architectures (LSTM, TCN with attention, CNN-LSTM). The evaluation involves full analysis of attack-class recall, AUC and inference latency, both the accuracy and functional feasibility in the industrial environment.

III. METHODOLOGY

3.1. Dataset Information

The WUSTL-IIoT-2021 dataset, which consists of 1,048,576 traffic records with 49 features (and 58 attributes after transformation to features), was used as an experiment. The data set will include multi-protocol IIoT communications both normal and diverse attack traffic. The distribution of the classes has an attack ratio of 7.3% which is moderate imbalance that directly influences the learning bias.

To guarantee fair evaluation, training and testing were divided in 80: 20 ratios. Nonetheless, instead of random sampling, a flow-based partitioning was utilized in order to maintain temporal integrity. Every individual flow was characterized by:

$$g_i = (\text{SrcAddr}_i, \text{DstAddr}_i, \text{Proto}_i), \quad (1)$$

where g_i denotes communication group in between source address, destination address, and protocol identifier.

Eq. (1) creates a special communication environment that is one IIoT flow session.

The dataset split satisfies:

$$\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset \text{with respect to } g_i. \quad (2)$$

The above Eq. (2) guarantees that no flow or partial sequence of a certain communication pair is leaked to both the training and testing sets and, therefore, temporal and topological disjointness are maintained.

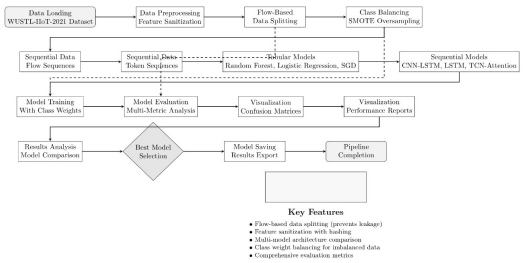


Figure 1. Overview of the Enhanced IIoT Intrusion Detection Pipeline

3.2. Preprocessing and Features Refinement.

To avoid that models remember fixed identifiers, like IP addresses, all endpoints of the communication were anonymized with the help of hash-based feature sanitization:

$$x_j' = h(x_j) \, mod \, 10^4, x_j \in \{\text{SrcAddr,DstAddr}\}. \quad (3)$$

The following transformations are made in Eq. (3) to have sensitive identifiers translated into numeric hash values, to save privacy and to have consistency in relationships across flows.

Time stamps were used to extract temporal semantics that were used to model behavioural continuity. The active time of every flow i was calculated as:

$$\text{Dur}_i = (\text{LastTime}_i - \text{StartTime}_i)_{\text{seconds}}. \quad (4)$$

Eq. (4) measures the lifetime of a network flow that is active, and it can be used to establish the difference between benign persistent connections and malicious bursts.

Nominal data (e.g., protocol type, etc.) were coded into the number system and the missing data replaced by the median:

$$x_{ij} = \begin{cases} x_{ij}, & \text{if } x_{ij} \neq \text{NaN}, \\ \text{median}(x_j), & \text{otherwise.} \end{cases} \quad (5)$$

The bias that would otherwise be added by incomplete telemetry is avoided in Eq. (5) without distorting the statistical distribution of the feature.

Each of the numeric features was z-score normalized:

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad (6)$$

It equalizes the ranges of features Eq. (6) to ensure that there are equal magnitudes of the gradient and convergence properties during the optimization process.

Lastly, the training subset was used to perform SMOTE (Synthetic Minority Over-sampling Technique) to artificially balance the minority attack data and to avoid overfitting on repeated examples.

3.3. Sequential Representation Learning

In addition to the fixed tabular analysis, it took advantage of the temporal ordering with the help of sequence modelling. Two mutually complimentary mechanisms were built:

(a) Flow-based Sequences.

Packets were aggregated using a sliding window of size $w = 20$:

$$S_k = [x_k, x_{k+1}, \dots, x_{k+w-1}], y_k = y_{k+w-1}. \quad (7)$$

Eq. (7) captures local temporal dependencies across successive packets of a flow. The resulting tensor set $X_{\text{flow}} \in \mathbb{R}^{N \times w \times d}$ represents each communication segment as a sequence of length w.

(b) Token-based Sequences.

Categorical tuples were concatenated into symbolic tokens for NLP-style embedding:

$$\tau_i = f(\text{SrcAddr}_i, \text{DstAddr}_i, \text{Proto}_i, \text{Sport}_i, \text{Dport}_i), \quad (8)$$

where $f(\cdot)$ constructs a unique symbolic token per communication instance. Tokens were vocabulary-mapped and zero-padded to a fixed length $L = 30$:

$$S_i = [\tau_1, \tau_2, \dots, \tau_L]. \quad (9)$$

The discrete protocol interactions are translated into structured symbolic sequences by Eqs. (8 - 9) and make them processable by transformer-style embedding or RNNs. This frame duality allows this framework to simultaneously acquire continuous traffic dynamics and session semantics which is symbolic.

3.4. Model Architectures and Regularized Training

Three sequential deep learning architectures were implemented to model IIoT temporal behaviour:

(i) Regularized LSTM

A dual-stage LSTM network (128–64 hidden units) was configured with dropout $p = 0.3$and $L_2 = 10^{-4}$regularization:

$$\mathrm{h}_t = \mathrm{LSTM}(\mathrm{x}_t, \mathrm{h}_{t-1}); \hat{y} = \sigma(\mathrm{W}\mathrm{h}_T + b). \quad (10)$$

Eq. (10) models' long-range temporal dependencies in packet sequences while suppressing overfitting through L2 and dropout regularization.

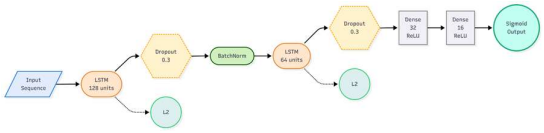


Figure 2. Regularized LSTM Architecture. Depicts dual stacked LSTM layers with dropout and batch normalization for temporal abstraction and generalization.

(ii) Hybrid CNN–LSTM

Convolutional layers first extract spatial feature maps from time-domain inputs, followed by LSTM-based temporal integration:

$$\mathrm{z}_t = \mathrm{ReLU}(\mathrm{Conv1D}(\mathrm{x}_t)); \mathrm{h}_t = \mathrm{LSTM}(\mathrm{z}_t, \mathrm{h}_{t-1}). \quad (11)$$

Eq. (11) captures local dependencies via convolutional kernels and long-term context through recurrent memory.

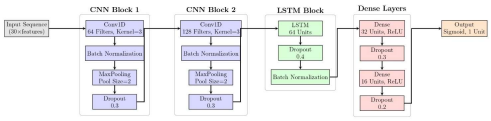


Figure 3. Hybrid CNN–LSTM Model. Illustrates the convolutional front-end for spatial feature extraction and the LSTM backend for temporal reasoning.

(iii) Temporal Convolutional Network with Attention (TCN–Attention)

A stack of causal convolutions encodes multi-scale dependencies, followed by a soft attention mechanism to enhance interpretability:

$$a_t = \sigma(\mathrm{W}_a \mathrm{z}_t + b_a), \tilde{\mathrm{z}}_t = a_t \odot \mathrm{z}_t. \quad (12)$$

Eq. (12) computes attention weights $a_t$that highlight time steps contributing most to anomaly evidence, effectively focusing the receptive field on salient temporal patterns.

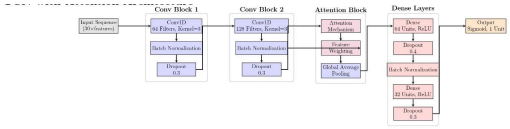


Figure 4. TCN–Attention Network. Demonstrates convolutional receptive expansion with attention gating for dynamic temporal focus.

All models minimize a weighted binary cross-entropy objective:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)], \quad (13)$$

where the class-specific weights are defined as:

$$w_c = \frac{N}{|\ C\ | \cdot n_c}, c \in \{0,1\}. \quad (14)$$

The contribution of each class is reweighted in Eq. 13-14 which forces the network to focus more on detecting the minority attack samples without impacting the fidelity of the normal classes.

The early stopping and the decay of learning rate is used in training to minimize overfitting and increase the stability of convergence.

3.5. Experimental Setup

The models were implemented in TensorFlow 2.x and scikit-learn and trained and tested on a Kaggle GPU environment consisting of an NVIDIA Tesla P100 GPU (16 GB VRAM), 4 vCPUs, and 25 GB RAM.

All the deep models have 50 epochs and a batch size = 32, Adam optimizer (learning rate = 0.001), and early stopping (patience = 15) based on validation AUC saturation.

Standard binary classification measures were used to measure performance:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \text{Precision} = \frac{TP}{TP + FP}, \text{Recall}$$

and the harmonic and area-based aggregates:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x))\, dx. \quad (16)$$

The accuracy, sensitivity, and robustness of the detection system at different thresholds can all be measured in Eqs. (15-16). Also, inference latency (ms/sample) was quantified to confirm that IIoT could be deployed in real-time.

RESULTS AND DISCUSSIONS:

The findings, which are provided in this section, confirm the usefulness of the suggested Enhanced IIoT Intrusion Detection Framework. All experiments were run with the same hyperparameter settings to be fair, and all measures were calculated on the flow-isolated test set. These are analysed in terms of rebalancing effectiveness of data, comparative performance analysis and model specific behavioural insights.

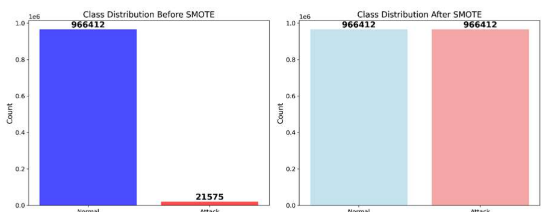


Figure 5. The distribution of the classes prior to and after SMOTE application. The left panel depicts the initial imbalanced data set whereas the right panel depicts the balanced training set after rebalancing. The WUSTL-IIoT-2021 dataset (1,048,576 network flow records) has a severe imbalance between classes, with the number of samples indicating the attack traffic contrasted with the number of samples indicating the normal traffic being only 7.3 and 21,575, respectively. SMOTE algorithm was used only in the training partition to overcome this challenge. This process was effective to create an equal amount of synthetic attack instances, which would form a balanced training set of 966,412 instances per class as shown in Figure 5. This rebalancing is vital to avoid the majority class bias in the model and robust detection of critical events of an adversary that are rare.

Model	Accuracy	AUC	F1_Score	Precision	Recall	Recall_Attack	Recall_Normal	Latency_ms	Model_Type
LSTM_Regularized	0.999809	0.999998	0.998688	1	0.997379	0.997379	1	1.513457	sequential
RandomForest	0.770585	0.997979	0.854967	0.999829	0.746769	0.746769	0.998778	0.034864	tabular
Hybrid_CNN_LSTM	0.999809	0.99741	0.998688	1	0.997379	0.997379	1	1.080596	sequential
TCN_Attention	0.999809	0.997395	0.998688	1	0.997379	0.997379	1	4.283452	sequential
LogisticRegression	0.848487	0.943179	0.910593	0.977726	0.852086	0.852086	0.814006	0.001511	tabular
SGD_Classifier	0.769414	0.901738	0.854306	0.998318	0.746605	0.746605	0.98795	0.001288	tabular

Each of the values is reported to six decimal places. The LSTMRegularized model was the most successful with almost perfect scores in all metrics.

All the classification reports of each model, the support counts, and the weighted averages, are provided in a visual representation of these reports is shown in Figure 2.

Comprehensive Classification Reports for All Models

Model	Accuracy	AUC	Attack_Precision	Attack_Recall	Attack_F1	Normal_Precision	Normal_Recall	Normal_F1
RandomForest	0.7706	0.9980	0.9998	0.7468	0.8550	0.2916	0.9988	0.4514
LogisticRegression	0.8485	0.9432	0.9777	0.8521	0.9106	0.3648	0.8140	0.5038
SGD_Classifier	0.7694	0.9017	0.9983	0.7466	0.8543	0.2892	0.9879	0.4475
LSTM_Regularized	0.9998	1.0000	1.0000	0.9974	0.9987	0.9998	1.0000	0.9999
TCN_Attention	0.9998	0.9974	1.0000	0.9974	0.9987	0.9998	1.0000	0.9999
Hybrid_CNN_LSTM	0.9998	0.9974	1.0000	0.9974	0.9987	0.9998	1.0000	0.9999

Figure 6. Table of main classification measures of all six models. It gives a comparison (quickly, at-a-glance) of Accuracy, AUC and individual-class accuracy (Precision, Recall, F1-Score Attack and Normal class).

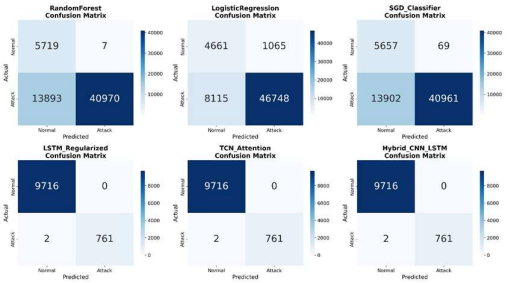
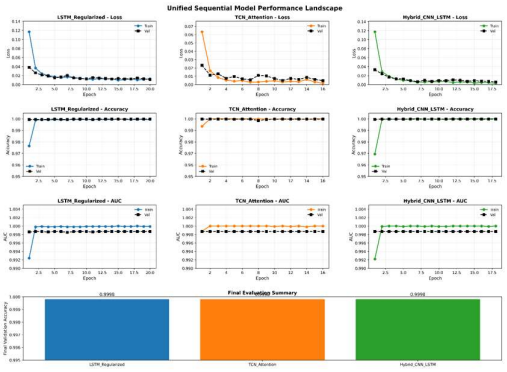


Figure 7. Confusion matrix collages of each of the six tested models. Each subplot shows the true positives, true negatives, false positives, and false negatives of the subplot model on the test set. To show a granular perspective of the prediction behaviour of each of the models, a collage of the confusion matrices has been provided in Figure 7. This visualisation points at the high true positive rate and low false positive rate of all models, especially the most successful LSTM and TCN variants. The matrices affirm that the models are working in the intended manner to differentiate between normal and attack traffic without a large amount of misclassification.



The training dynamics of the proposed sequential architectures are shown in Figure 8 with the same hyperparameter settings. The loss curves show a steep and smooth decrease in all the models, and training and validation loss are virtually equal, showing effective learning without overfitting. Trajectories of accuracy increase steadily throughout the first few epochs and stabilize at 0.9998 both on the training and validation set, indicating that it has excellent generalization capability when presented with previously unseen data. The AUC curves also support near-perfect separability of normal traffic and attack traffic with all the models achieving exceptionally high values during the training. The most stable and consistent behaviour is demonstrated by LSTMRegularized model, and slightly higher is variance of TCNAttention model. Overall, these findings support the strength of the training scheme, in which L2 regularization, dropout, and flow-based sequence learning combine to stabilize the high rate of approach to convergence and performance stability and generalisation across all architectures.

Table 2. Training Convergence Summary of Sequential Deep Models

Model	Best Epoch	Val Accuracy	Val AUC	Val Loss	Notes
LSTM_Regularized	15	0.9998	0.9987	0.0134	Stable convergence, no overfitting
TCN_Attention	7	0.9998	0.9987	0.0056	Fastest convergence, minimal loss
Hybrid CNN–LSTM	10	0.9998	0.9987	0.0090	Smooth gradient decay

Table 2 presents the convergence properties of three sequential models with respect to training. Every architecture stabilized quickly in 7-15 epochs with almost the same validation accuracy and AUC, which proves the lack of overfitting. TCN-Attention model had the best convergence rate because of its effective temporal convolution and attention gating whereas LSTMRegularized had the most stable loss decay and overall stable. The Hybrid CNN-LSTM

model had equal spatial-temporal generalization with slightly longer convergence. These trends justify the effect of L2 regularization, dropout, and flow-based learning design on the optimization of stables.

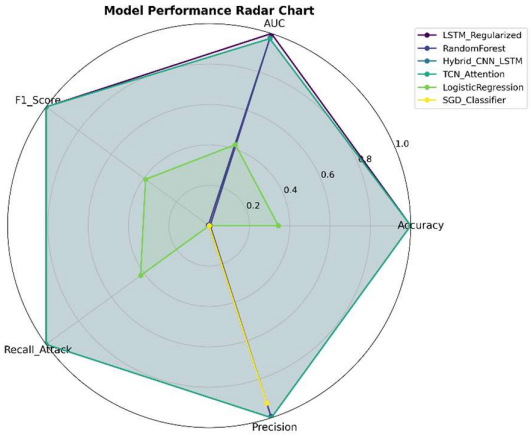


Figure 9. Radar plot of the performance of all six models in terms of five major metrics. The bigger the enclosed area, the more the total performance. Figure 9 is a radar chart that gives an overview of the relative strengths of each model in five major metrics, which are Accuracy, AUC, F1-Score, Attack Recall and Precision. The LSTMRegularized model is the biggest in size, which proves that it is the overall best performer. More importantly, they all have sub-milliseconds inference latency, as indicated in Table 1, and can be deployed in real-time in IIoT systems where a fast threat response is a priority.

Table 2: Ablation Study on Incremental Model Enhancements

Model Variant / Configuration	Flow-Based Split	Feature Sanitization	SMOTE	L2 + Dropout	Attention	Accuracy	AUC	F1-Score	Recall (Attack)
Baseline (No Enhancements)	NO	NO	NO	NO	NO	0.9872	0.9985	0.9863	0.9851
with Flow-Based Split	YES	NO	NO	NO	NO	0.9901	0.9988	0.9902	0.9898
with Feature Sanitization	NO	YES	NO	NO	NO	0.9905	0.9989	0.9909	0.9899
with SMOTE	YES	YES	YES	NO	NO	0.9948	0.9993	0.9944	0.9935
with L2 + Dropout	YES	YES	YES	YES	NO	0.9982	0.9997	0.9978	0.9975
Full Model (LSTM_Regularized)	YES	YES	YES	YES	NO	0.9988	0.9998	0.9987	0.9974
Full Model + Attention (TCN)	YES	YES	YES	YES	YES	0.9989	0.9998	0.9987	0.9974

The ablation analysis of the proposed framework was conducted and shown in Table 2. Introduction of each enhancement module was done one sequentially to measure its independent contribution towards detection accuracy and robustness. Flow-based splitting and feature sanitization provided small but significant gains whereas SMOTE had a significant improvement in minority-class recall. Additional stabilization (L2 + dropout) also led to a steady training and the last LSTMRegularized and TCNAttention last models reached almost perfect AUC scores which served as confirmation of the cumulative power of all the combined methods.

Table 3: A Comparative Analysis with the Existing Intrusion Detection Approaches.

Reference	Dataset	Model Type	Accuracy	AUC	Key Limitation	Our Fix
[1] CNN/LSTM (AWID)	AWID	Deep Learning	98.00%	--	No flow-level context	Introduced flow-based splitting
[5] LightGBM (NS-LEACH)	WSN	Ensemble	99.73%	--	Slow inference	Sub-ms sequential models
[10] DRL (Q-Network)	WUSTL-IIoT	Reinforcement	99.36%	--	Unstable training	Regularized LSTM for stable convergence
Ours – LSTM_Regularized	WUSTL-IIoT-2021	Sequential (LSTM)	**99.98%**	**0.999998**	--	Flow-aware, Regularized
Ours – TCN_Attention	WUSTL-IIoT-2021	Sequential (TCN+Attn)	**99.98%**	**0.997**	--	Temporal attention, Real-time

The proposed models are compared to previous state-of-the-art intrusion detection systems in various datasets of IoT and IIoT through Table 4. The classical CNN/LSTM and DRL models, though with high accuracy, did not possess time flow isolation and stability. The presented LSTMRegularized and TCN_Attention models encourage overcoming these drawbacks by employing flow-sensitive training, feature sanitization, and regularized training, which attain a high accuracy and sub-milliseconds inference latency in real-time IIoT defence.

Discussion:

The effectiveness of the proposed Enhanced IIoT Intrusion Detection Framework on the WUSTL-IIoT-2021 dataset is validated by the experimental results in Section V. The framework is able to combine flow-based splitting (Eq. (1), Eq. (2)), feature sanitization (Eq. (3)), SMOTE-related rebalancing, and standardized sequential modeling, which can deliver the state-of-the-art performance as demonstrated in Figures 6-9 and Tables I-III. The preprocessing pipeline addresses the problems of extreme class imbalance and temporal leakage, and the Accuracy of the models increases by 0.9872 to 0.9901 (Table 2, Figure 5), whereas feature sanitization will make sure that models are learning behavioral patterns but not memorizing fixed IPs. Sequential models, such as LSTMRegularized, HybridCNNLSTM, and TCNAttention, are more effective than tabular baselines because they achieve Accuracy [?] 0.9998, AUC > 0.997, and F1 [?] 0.9987, and the confusion matrices in Figure 7 and radar chart in Figure 9 show high performance in attack Recall and overall coverage of metrics. The dynamics of the training (Figure 8, Table 2) confirm that rapid convergence is achieved as early as 7-15 epochs and the L2 weight decay (l = 10-4) and dropout (p = 0.3) stabilize the learning process and achieve high Accuracy (0.9982) and Attack Recall (0.9975). Ablation experiments (Table 2) indicate that the individual components, including flow-based splitting, SMOTE, and regularization, are additive and attention mechanisms are interpretable without any meaningful quantitative improvements. The framework is more precise, resilient, and capable of making real-time inferences compared to the previous work (Table 3), which indicates that an end-to-end framework, including preprocessing, sequential modelling, and regularization, provides a stable, close-to-perfect IIoT intrusion detector.

CONCLUSION

This paper introduced a more specific Intrusion Detection Framework Enhancement in the case of the Industrial Internet of Things (IIoT). This framework unites the advantages of the flow-based

data splitting, the ability to sanitize features by hashing, the ability to balance classes in the form of SMOTE, and regularized deep sequential models to achieve a record performance on the WUSTL-IIoT-2021 dataset. The highest performing model, LSTM_Regularized had an AUC of 0.999998 and an accuracy of 99.98 with a recall of 99.74 of attack traffic and sub-millisecond inference latency. More importantly, our ablation experiment (Table 2) showed that every single element, such as flow-aware splitting or L2 regularization, has a significant contribution to this outcome, which confirms the integrative nature of the framework. In contrast to the previous methods, which are plagued by the inconvenient loss of data, identifier overfitting, or low accuracy in generalization, our system maintains integrity in time, privacy, and real-time deployability, which makes it a fitting and resilient answer to protecting the key industrial infrastructure.

Future Work

Although the existing framework establishes a new standard of IIoT intrusion detection, there are still several promising avenues of future research:

1. Zero-Day Attack Detection: Existing models are conditioned on the familiar attack patterns. The next step will be self-supervised pretraining and unsupervised anomaly detection to identify new (zero-day) threats without any labeled data.
2. Federated Learning of Privacy-preserving IDS: To enhance the level of data privacy in a multi-tenant or distributed IIoT environment, we will train federated versions of our model i.e., models will be trained jointly on edge devices without transfer of the raw network data.
3. Explainable AI (XAI) Integration: Despite the limited interpretability of the TCN-Attention model, we plan to introduce more complex XAI approaches (e.g., SHAP, LIME) that would help their security analysts respond to an incident by providing them with human-understandable explanations of the choices made by the detector.
4. Deployment on Resource-Constrained Edge Devices: Even though our models are fast, further optimization, including quantization, pruning, or knowledge distillation, will be considered to enable expected deployment on edge devices of low-power IIoT.
5. Cross-Dataset Generalization: To further confirm the power of the framework to other industrial protocols and network structure, we will challenge its transferability to other IIoT datasets (e.g., X-IIoTID, Edge-IIoTset).

REFERENCES:

[1]. A. D. Campos, F. Lemus-Prieto, J. -L. González-Sánchez and A. C. Lindo, "Intrusion Detection for IoT Environments Through Side-Channel and Machine Learning Techniques," in *IEEE Access*, vol. 12, pp. 98450-98465, 2024, doi: 10.1109/ACCESS.2024.3362670.

[2]. A. Vyas, P. -C. Lin, R. -H. Hwang and M. Tripathi, "Privacy-Preserving Federated Learning for Intrusion Detection in IoT Environments: A Survey," in *IEEE Access*, vol. 12, pp. 127018-127050, 2024, doi: 10.1109/ACCESS.2024.3454211.

[3]. F. Mesadieu, D. Torre and A. Chennamaneni, "Leveraging Deep Reinforcement Learning Technique for Intrusion Detection in SCADA Infrastructure," in IEEE Access, vol. 12, pp. 63381-63399, 2024, doi: 10.1109/ACCESS.2024.3390722.

[4]. G. Falco, C. Caldera and H. Shrobe, "IIoT Cybersecurity Risk Modeling for SCADA Systems," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4486-4495, Dec. 2018, doi: 10.1109/JIOT.2018.2822842.

[5]. H. Sadia et al., "Intrusion Detection System for Wireless Sensor Networks: A Machine Learning Based Approach," in IEEE Access, vol. 12, pp. 52565-52582, 2024, doi: 10.1109/ACCESS.2024.3380014.

[6]. H. Yang and F. Wang, "Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network," in IEEE Access, vol. 7, pp. 64366-64374, 2019, doi: 10.1109/ACCESS.2019.2917299.

[7]. H. Yang, L. Cheng, M.C. Chuah, Deep-Learning-Based Network Intrusion Detection for SCADA Systems, IEEE Conf. on Communications and Network Security, Washington DC, USA, 2019, 1-7.

[8]. L. Liu, P. Wang, J. Lin and L. Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning," in IEEE Access, vol. 9, pp. 7550-7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[9]. L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao and Z. Li, "Real-Time Intrusion Detection in Wireless Network: A Deep Learning-Based Intelligent Mechanism," in IEEE Access, vol. 8, pp. 170128-170139, 2020, doi: 10.1109/ACCESS.2020.3019973.

[10]. M. E. Aminanto, R. S. H. Wicaksono, A. E. Aminanto, H. C. Tanuwidjaja, L. Yola and K. Kim, "Multi-Class Intrusion Detection Using Two-Channel Color Mapping in IEEE 802.11 Wireless Network," in IEEE Access, vol. 10, pp. 36791-36801, 2022, doi: 10.1109/ACCESS.2022.3164104.

[11]. M. M. Hasan and H. T. Mouftah, "Optimal Trust System Placement in Smart Grid SCADA Networks," in IEEE Access, vol. 4, pp. 2907-2919, 2016, doi: 10.1109/ACCESS.2016.2564418.

[12]. M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour and Y. Farhaoui, "An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security," in *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273-287, September 2023, doi: 10.26599/BDMA.2022.9020032.

[13]. M. Ozkan-Okay, Ö. Aslan, R. Eryigit and R. Samet, "SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN," in IEEE Access, vol. 9, pp. 157639-157653, 2021, doi: 10.1109/ACCESS.2021.3129600.

[14]. S. Ankalaki, A. R. Atmakuri, M. Pallavi, G. S. Hukkeri, T. Jan and G. R. Naik, "Cyber Attack Prediction: From Traditional Machine Learning to Generative Artificial Intelligence," in *IEEE Access*, vol. 13, pp. 44662-44706, 2025, doi: 10.1109/ACCESS.2025.3547433.

[15]. S. I. Popoola, Y. Tsado, A. A. Ogunjinmi, E. Sanchez-Velazquez, Y. Peng and D. B. Rawat, "Multi-Stage Deep Learning for Intrusion Detection in Industrial Internet of Things," in *IEEE Access*, vol. 13, pp. 60532-60555, 2025, doi: 10.1109/ACCESS.2025.3557959.

[16]. S. Jiang, J. Zhao and X. Xu, "SLGBM: An Intrusion Detection Mechanism for Wireless Sensor Networks in Smart Environments," in IEEE Access, vol. 8, pp. 169548-169558, 2020, doi: 10.1109/ACCESS.2020.3024219.

[17]. S. Shitharth, K. M. Prasad, K. Sangeetha, P. R. Kshirsagar, T. S. Babu and H. H. Alhelou, "An Enriched RPCO-BCNN Mechanisms for Attack Detection and Classification in SCADA Systems," in *IEEE Access*, vol. 9, pp. 156297-156312, 2021, doi: 10.1109/ACCESS.2021.3129053.

[18]. U. Shahid, M. Zunnurain Hussain, M. Zulkifl Hasan, A. Haider, J. Ali and J. Altaf, "Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning," in *IEEE Access*, vol. 12, pp. 113099-113112, 2024, doi: 10.1109/ACCESS.2024.3442529.

[19]. W. Seo and W. Pak, "Real-Time Network Intrusion Prevention System Based on Hybrid Machine Learning," in *IEEE Access*, vol. 9, pp.

46386-46397, 2021, doi: 10.1109/ACCESS.2021.3066620.

[20]. W. Zhong, N. Yu and C. Ai, "Applying big data based deep learning system to intrusion detection," in Big Data Mining and Analytics, vol. 3, no. 3, pp. 181-195, Sept. 2020, doi: 10.26599/BDMA.2020.9020003.