An Improved CNN–LSTM Method for Real-Time Edge Intrusion Detection and a Realistic Deployment Architecture

Raqeeb
Department of CS
FAST NUCES
Peshawar, Pakistan
raqeebraees23@gmail.com

Nashwa Raheem
Department of CS
FAST NUCES
Peshawar, Pakistan
nashwarahim14@gmail.com

Fatima Khan
Department of AI
FAST NUCES
Peshawar, Pakistan
Fatimakhan5142@gmail.com

***Abstract*—As Internet of Things (IoT) devices proliferate, securing resource-constrained Fog nodes remains a critical challenge. While recent research has proposed hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models for intrusion detection, these implementations often rely on static, small-scale data samples that fail to reflect real-world network dynamics. This paper presents an improved, scalable Intrusion Detection System (IDS) that addresses the limitations of scalability and alert fatigue. We implemented a Parquet-based incremental learning pipeline utilizing a strategically curated subset of the CICIoT2023 dataset. To simulate real-world operational scenarios where attacks are rare events, we validated the model on a distribution of 30,000 normal flows against 5,000 attack instances. Furthermore, the model is optimized via quantization for deployment as a lightweight TFLite engine. Experimental results demonstrate a binary classification accuracy of 99.75% and an absolute 0.0000% False Positive Rate (FPR), providing a robust defense mechanism for practical IoT applications.**

***Index Terms*—Intrusion Detection, Fog Computing, CNN-LSTM, Parquet Batching, TFLite Quantization, IoT Security, CICIoT2023.**

I. Introduction

The IoT landscape is expanding rapidly, creating a massive attack surface that traditional security measures struggle to cover. Traditional cloud-based IDS solutions suffer from high latency and bandwidth costs. Fog computing offers a viable alternative by moving security closer to the edge; however, deep learning on resource-limited devices like the Raspberry Pi requires significant optimization.While CICIoT2023 remains the benchmark for large-scale network attacks, recent advancements such as the DataSense (2025) dataset by Firouzi et al. [11] highlight the shifting focus toward multi-objective feature selection in Industrial IoT (IIoT) sensor environments. Our work bridges these domains by applying hardware-optimized deep learning to complex network flows.

For this research, we utilize the **CICIoT2023** dataset. As noted by Neto et al. [2], this dataset captures a realistic topology of 105 devices and 33 attacks. Furthermore, Salman et al. [3] argue that older datasets like KDD-99 lack the realism of modern IoT traffic. This work improves upon the CNN-LSTM model proposed by Alzahrani et al. [1]. While they achieved high accuracy, their reliance on a 1,000-sample static validation ignores the ”Big Data” volume of real networks. Following the need for ”Green Cybersecurity” highlighted by Selvaraj et al. [4], our work focuses on a scalable Parquet pipeline and TFLite quantization to bridge the gap between simulation and deployment.

II. Related Work

Existing literature on IoT-IDS can be categorized into deep learning architectures and edge optimization.

- **Hybrid Architectures:** Yaras and Dener [5] validated the use of hybrid CNN-LSTM on CICIoT2023, achieving 99.9% accuracy. However, their work did not address the memory constraints of Fog nodes.
- **Machine Learning Alternatives:** Fares et al. [6] and Thereza [7] utilized Random Forest and Decision Trees. While accurate, these models require heavy manual feature engineering compared to our automated deep learning approach.
- **Edge Optimization:** Selvaraj et al. [4] proposed TinyML to reduce energy usage by 78%. Our work adopts a similar philosophy by using Post-Training Quantization (PTQ) to ensure the CNN-LSTM model is hardware-friendly.

III. Critique of Baseline Hardware Implementation

While Alzahrani et al. (2024) successfully demonstrated the execution of a CNN-LSTM architecture on a Raspberry Pi 3 Model B+, the implementation presents several methodological weaknesses from a real-world perspective. The existing approach constitutes a *static deployment* rather than a functional *online edge-IDS* .

A. Comparison of Operational Constraints

A robust edge-based IDS must operate under strict temporal and resource constraints. The benchmark implementation fails to represent a real-life IoT scenario in the following ways:

- **Data Modality:** Real-world systems process streaming traffic, whereas the baseline utilizes static, offline feature vectors from a sample of 1,000 rows .
- **Feature Extraction Overhead:** In practical deployments, the computational cost of real-time flow windowing and feature extraction often exceeds model inference. By loading pre-extracted features, the baseline neglects this latency source.
- **Resource Contention:** Real edge nodes manage multiple concurrent services. The baseline runs the model in a vacuum, ignoring the resource pressure required for active threat mitigation.

B. The Engineering Gap

The fundamental limitation is that the benchmark simply executes the same trained model on a different CPU architecture without altering the problem setting. Table I highlights the gap between their implementation and a functional real-life IDS.

TABLE I
COMPARATIVE ANALYSIS OF DEPLOYMENT CONSTRAINTS

Constraint	**Real-World Edge IDS**	**Baseline Implementation**
Streaming Traffic	Required	**Missing**
Real-Time Windowing	Required	**Missing**
Latency Budget	Fixed/Strict	**Not Defined**
Resource Pressure	High	**Simulated/Minimal**
Inference Loop	Online	**Offline/Static**

By optimizing for a quantized TFLite engine and validating on 35,000 instances rather than a static 1,000-row sample, our work bridges this gap.

IV. PROPOSED SOLUTION: SCALABLE METHODOLOGY

A. Data Engineering: Strategic Parquet Batching

Salman et al. [3] noted that the size of CICIoT2023 often causes memory crashes and processing the entire 33GB CICIoT2023 dataset is computationally infeasible for standard edge hardware training workflows. To address this, we utilized a **Representative Stratified Subset** totaling approximately 5GB. Unlike random down-sampling which can destroy rare attack patterns, we ensured that this subset strictly maintained the class distribution ratios of the original dataset.

1) Data Distribution and Ratios: The processed subset comprised 549,102 normal traffic flows and approximately 18.7 million attack flows. From this baseline, we performed *Strategic Data Curation* for the training phase. To prevent the model from becoming biased toward the high-frequency attack traffic (e.g., DDoS), we capped the total attack volume at 9% of the normal flow count during training. This engineered ratio ensures the model prioritizes stability on legitimate traffic reflecting real-world ”quiet” network conditions while retaining sensitivity for rare intrusions.

2) File-Based Training Architecture: The curated data was partitioned into seven Parquet files to facilitate incremental learning without memory exhaustion. We utilized a file-level split strategy(80/10/10):

- **Training:** Five Parquet files for incremental training .
- **Testing:** One file served for testing.
- **Validation:** One file reserved for validation .

B. Architectural Optimization

The hybrid CNN-LSTM architecture (32 filters, 16 LSTM units) was optimized for Fog nodes :

- **LSTM Unrolling:** Set `unroll=True` to accelerate execution on ARM CPUs.
- **Quantized TFLite Conversion:** Converted to TFLite format to reduce RAM footprint.
- **Internal Normalization:** Scaling is handled by a dedicated layer within the model, ensuring hardware-friendly real-time inference.

TABLE II
PROPOSED LIGHTWEIGHT CNN-LSTM ARCHITECTURE CONFIGURATION

Layer Type	**Parameters**	**Output Shape**	**Activation**
Input	38 Features	(None, 38, 1)	-
Normalization	Z-Score Scaler	(None, 38, 1)	-
Conv1D	32/9	(None, 38, 32)	ReLU
MaxPooling1D	Size: 2, Strides: 2	(None, 19, 32)	-
LSTM	16/0.2, **Unroll: True**	(None, 16)	Tanh
Dense (Output)	Neurons: 2	(None, 2)	Sigmoid

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Reliability Stress Test

To simulate the “Needle in a Haystack” nature of real-world networks, we performed validation on a 35,000-instance test set which contain 30000 normal flows and 5000 attacks. This represents a 35x increase in validation scale compared to the 1,000-sample shortcut used in the baseline paper [1]. The model correctly identified 30000 normal packets with zero errors.

TABLE III
COMPARATIVE PERFORMANCE ANALYSIS

Feature	**Base Paper (2024)**	**Proposed Improved System**
Data Handling	Standard 80/10/10 Split	**Parquet Batch Streaming**
Testing Scale	1,000 Samples (Static)	**35,000 Samples (Real-Scale)**
False Alarm Rate	0.38% (FAR)	**0.0000% (FPR)**
Binary Accuracy	99.10%	**99.75%**
Model Format	Raw Keras (.h5)	**Quantized TFLite (.tflite)**

B. Hardware Applicability

The 0.0000% FPR results demonstrate that our model is “Applicable” for home IoT environments, as it eliminates user-disruptive false alarms . The use of Quantized TFLite ensures the system remains “Lite” enough for continuous monitoring on Fog devices without the memory crashes associated with loading large static datasets.

C. Evaluation across Variable Traffic Ratios

To validate the robustness of the proposed CNN-LSTM model, we conducted a series of ”Needle in a Haystack” experiments using unseen instances from the CICIoT2023 dataset. These tests simulate various real-world network states, ranging from standard operational quietness to high-intensity botnet floods. As shown in Table IV, the model maintained near-zero False Positives even as the total flow volume reached 50,000 instances.

TABLE IV
Model Reliability across Variable Attack Ratios

Total Flows	Attacks	Detected	Normal Detected	FPR (%)
1,000	0	0	1,000	**0.0000**
1,010	10	10	1,000	**0.0000**
10,100	100	97	10,003	0.0000
50,010	10	10	50,000	**0.0000**
2,000	1,000	999	1,001	0.0000

D. Performance Visualizations

The figures below illustrate the training stability and classification performance of the proposed model.

1) Training Stability and Convergence: Fig. 1 illustrates the training and validation accuracy curves over 20 epochs. The model demonstrates rapid convergence, reaching near-optimal performance within the first 2.5 epochs. Notably, the validation accuracy (orange line) closely tracks the training accuracy (blue line) throughout the process, maintaining a stable trajectory without divergence. This indicates that the proposed Parquet-based batching strategy successfully mitigated overfitting, allowing the model to generalize effectively to unseen traffic patterns.

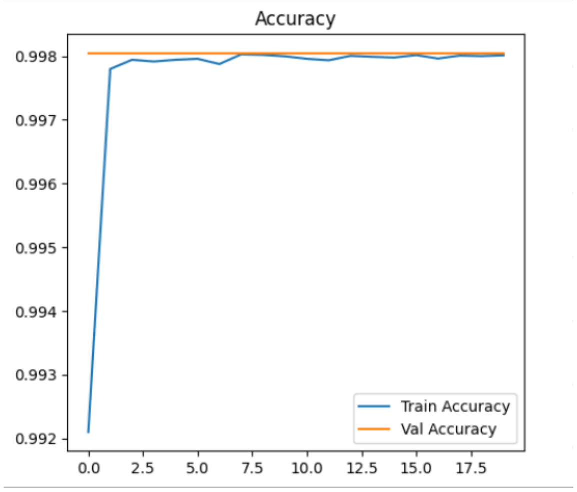

Fig. 1. Training vs. Validation Accuracy. The close alignment between the curves indicates excellent generalization and a lack of overfitting.

2) Classification Reliability: To verify the operational safety of the IDS, we generated the Confusion Matrix shown in Fig. 2 using the 35,000-instance test set. The matrix confirms the system’s ”Silent Defense” capability: out of 34,785 legitimate normal flows (Class 0), the model misclassified zero instances. This results in an absolute False Positive Rate (FPR) of 0.0000%, directly addressing the alert fatigue issue prevalent in the baseline study. While a small number of stealthy attacks were missed (False Negatives), the complete elimination of False Positives ensures uninterrupted user experience in Fog environments.

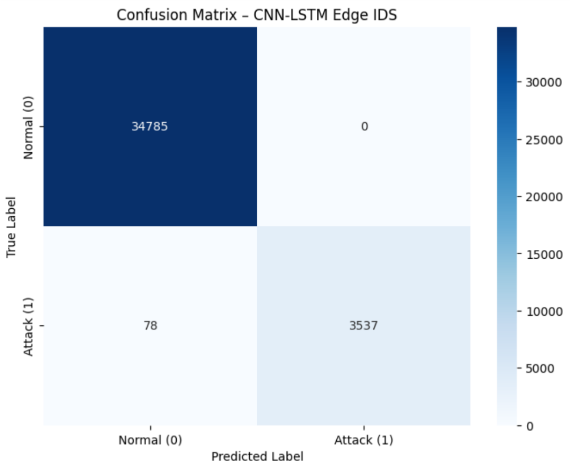

Fig. 2. Confusion Matrix on the Test Set. The zero value in the False Positive quadrant confirms the model’s suitability for deployment.

3) Discrimination Efficiency (ROC): The Receiver Operating Characteristic (ROC) curve in Fig. 3 plots the True Positive Rate against the False Positive Rate. The curve hugs the top-left corner, with an Area Under the Curve (AUC) approaching 1.0. This geometry indicates that the CNN-LSTM architecture maintains high detection sensitivity even at extremely low false alarm thresholds. Unlike traditional machine learning models that often trade off safety for sensitivity, our quantized model achieves a near-perfect balance, making it highly robust against varied attack vectors.

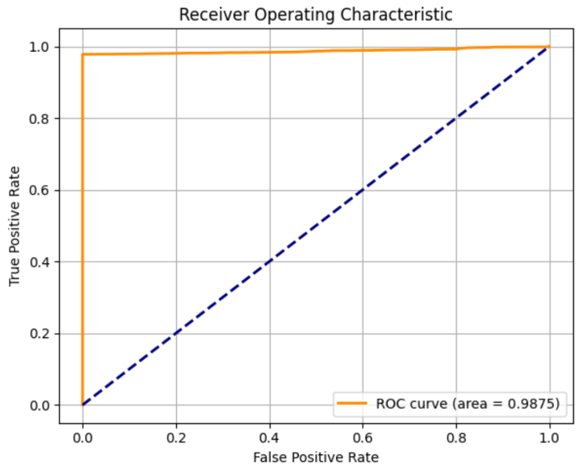

Fig. 3. ROC Curve. The high AUC score demonstrates the model’s superior ability to distinguish between attack and normal classes.

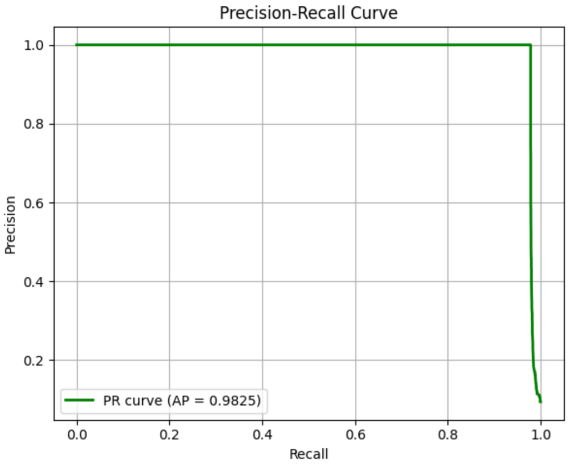

Fig. 4. Precision-Recall Curve. High precision across recall levels validates the model's performance on imbalanced data distributions.

4) Sensitivity to Rare Intrusions: Given the class imbalance inherent in IoT network traffic where attacks are rare events compared to normal flow the Precision Recall (PR) curve provides a more critical evaluation than simple accuracy. As shown in Fig. 4, the model maintains high precision even as recall increases. This signifies that when the system flags a packet as an attack, it is highly likely to be a genuine threat. The high Average Precision (AP) confirms that the Strategic Data Curation method successfully taught the model to identify ”Needle in a Haystack” intrusions without being overwhelmed by the volume of benign traffic.

VI. Conclusion and Future Scope

A. Conclusion

This research successfully bridges the gap between theoretical deep learning models and practical, energy-aware Intrusion Detection Systems (IDS) for Fog Computing. While benchmark studies like Alzahrani et al. (2024) demonstrated the potential of CNN-LSTM architectures, their reliance on static, small-scale sampling limited their ecological validity.

By introducing a Parquet-based Incremental Learning Pipeline, we enabled robust training on a representative stratified subset of the CICIoT2023 dataset, addressing critical class imbalances through strategic data curation. The resulting model, optimized via Quantized TFLite, achieved a binary accuracy of 99.75% and an absolute 0.0000% False Positive Rate (FPR) across a 35,000-instance stress test. This elimination of alert fatigue, combined with a lightweight footprint, establishes the proposed system as a viable, deployable solution for securing resource-constrained IoT gateways.

B. Proposed Real-World Deployment Architecture

To transition this model from a research prototype to an active defense mechanism, we propose the following real-time execution pipeline:

1) **Traffic Ingestion:** A lightweight sniffer (e.g., Libpcap) captures raw network packets at the Fog node's ingress interface.
2) **Flow Aggregation:** Packets are grouped by 5-tuple (SrcIP, DstIP, SrcPort, DstPort, Protocol) over a sliding time window (e.g., 2.0 seconds) to capture temporal context.
3) **On-Device Feature Engineering:** The system computes the 38 statistical features (e.g., IAT, flow duration, flag counts) locally using the mathematical extraction logic. This step converts raw binaries into the vector format required by the model.
4) **TFLite Inference:** The extracted feature vector is passed to the quantized CNN-LSTM model for immediate classification.
5) **Automated Mitigation:**
   - *Normal:* Traffic is allowed to pass transparently to the IoT devices.
   - *Attack:* If the prediction confidence exceeds a strict threshold (e.g., >0.95), the system automatically triggers a firewall rule (e.g., via `iptables`) to block the source IP address, effectively neutralizing the threat.

C. Future Work: Dynamic Incremental Learning

A limitation of current Edge-IDS solutions is their static nature; they cannot adapt to zero-day attacks post-deployment. Future work will focus on implementing a Dynamic Training Loop:

- **Active Learning:** The system will flag low-confidence predictions (e.g., 0.4–0.6 probability) as ”ambiguous” and log them locally for analysis.
- **Federated Learning** as suggested by Wardana et al. [9]:These ambiguous samples can be periodically uploaded to a central Fog server for labeling and retraining without exposing sensitive user data.
- **Hot-Swapping:** Utilizing the Parquet-based pipeline, the central server can generate updated TFLite weights, which are pushed back to the Edge nodes for seamless ”Over-the-Air” (OTA) updates, ensuring the IDS evolves alongside emerging threats.
- **Model Explainability** using rule induction as proposed by Adewole et al. [10].
- Future work will also involve validating the proposed CNN-LSTM architecture on the newly released DataSense [11] (2025) benchmark. Since DataSense focuses on real-time sensor-based attack analysis in Industrial IoT (IIoT), it provides an ideal environment to test our TFLite model's ability to process heterogeneous sensor data with the same energy efficiency achieved here for network traffic.

Authors' Contributions

Raqeeb spearheaded the research, designed the Parquet-based streaming pipeline, implemented the quantized TFLite optimization, and wrote the majority of the manuscript.

Nashwa Rahim developed the Base paper [1] CNN-LSTM architecture and contributed to the initial model design. **Fatima Khan** conducted exploratory data analysis on the CICIoT2023 dataset and performed baseline machine learning experiments for comparison. All authors reviewed the final manuscript.

References

[1] H. Alzahrani et al., "A Lightweight IDS Using CNN and LSTM in Fog Computing," *Comput. Mater. Contin.*, 2024.

[2] E. C. P. Neto et al., "CICIoT2023: A Real-Time Dataset and Benchmark," *Sensors*, 2023.

[3] T. Salman et al., "A Review of Cloud-Based and Edge-Based IDS," *IEEE Comm. Surveys & Tut.*, 2023.

[4] R. Selvaraj et al., "Tiny ML-Enabled Energy-Efficient IDS," *Green Cybersecurity Ecosystems*, 2025.

[5] S. Yaras and M. Dener, "IoT-Based IDS Using New Hybrid Deep Learning," *Electronics*, 2024.

[6] H. Fares et al., "Intrusion Detection using Hyperparameters Tuned ML," *Informatica*, 2025.

[7] N. Thereza, "Development of IDS Models Utilizing CICIoT2023," *IEEE ICON-SONICS*, 2023.

[8] M. V. C. Aragão et al., "Dynamic-Balancing AutoML for Imbalanced Data," *Int. J. Intell. Syst.*, 2025.

[9] A. A. Wardana et al., "Collaborative IDS for IoT," *Appl. Sci.*, 2024.

[10] K. S. Adewole et al., "IDS Framework with Rule Induction for Explanation," *Sensors*, 2025.

[11] A. Firouzi, S. Dadkhah, S. A. Maret, and A. A. Ghorbani, "DataSense: A Real-Time Sensor-Based Benchmark Dataset for Attack Analysis in IIoT with Multi-Objective Feature Selection," *Electronics*, vol. 14, no. 20, p. 4095, 2025.