RESEARCH ARTICLE

TinyML Driven Intrusion Detection for 5G Network Slices with Leakage-Free Validation

Phalguni Patnaik[1], **Susrita Mishra**[1], **Bandhan Panda**[1,*] **and Santosh Kumar Kar**[1]

[1] Computer Science and Engineering, NIST University, Berhampur 761008, India

Abstract

The intrusion detection at the 5G network perimeter demands learning frameworks that are practically feasible and computationally efficient. This research proposes a lightweight, slice-sensitive intrusion detection approach designed for edge deployment, with a strong emphasis on minimizing information leakage while accounting for the resource constraints inherent in edge environments. A rigorous chronological and session-discontinuous experimental protocol ensures that training and test traffic remain temporally separated, faithfully replicating realistic deployment conditions. The proposed framework employs a classical Logistic Regression classifier using flow-based statistical features extracted from the 5G-NIDD dataset. To reduce model complexity while preserving detection performance, feature importance-based pruning is applied to retain only the most informative features, followed by post-training INT8 quantization. Rather than focusing on hardware-specific implementations, edge feasibility is assessed through software-based metrics, including model size, computational cost per inference, and CPU inference latency. Experimental results demonstrate that the optimized model exhibits stable intrusion detection performance under leakage-free conditions, achieving results largely comparable to—and in some cases slightly superior to—the full-feature baseline. Notable improvements in memory footprint and computational overhead are achieved, resulting in inference latencies of less than one millisecond in software simulations. Slice-wise analysis reveals predictable and interpretable behavior for both enhanced Mobile Broadband (eMBB) and massive Machine-Type Communications (mMTC) traffic, while conclusions regarding Ultra-Reliable Low-Latency Communications (URLLC) traffic are drawn cautiously due to insufficient representation in the dataset. These findings suggest that carefully constrained classical models, combined with feature-based optimization and strict evaluation protocols, provide a practical and transparent foundation for slice-aware intrusion detection at the 5G edge.

Keywords: 5G network intrusion detection, edge-based ID, TinyML for network security, leakage-free IDS evaluation, feature-pruned machine learning, 5G network slicing security.

Submitted: 16 January 2026
Accepted: 17 March 2026
Published: 19 March 2026

Vol. 2, **No.** 1, 2026.
10.62762/NGCST.2026.664893

***Corresponding author:**
✉ Bandhan Panda
bandhan.panda@nist.edu

Citation
Patnaik, P., Mishra, S., Panda, B., & Kar, S. K. (2026). TinyML Driven Intrusion Detection for 5G Network Slices with Leakage-Free Validation. *Next-Generation Computing Systems and Technologies*, 2(1), 10–20.

1 Introduction

The fifth-generation (5G) mobile network provides architectural flexibility and performance increase never before seen due to virtualization, softwarization, and network slicing, but the same features increase the attack surface by a large margin. With the growing use of heterogeneous networks based on 5G networks to offer heterogeneous services that may have varying quality-of-service demands, the ability to detect intrusions in a robust and timely way has become a critical security requirement. Intrusion Detection Systems (IDS) have continued to be one of the most fundamental mechanisms in terms of detecting malicious actions that would circumvent preventive security measures.

Recent research has addressed how machine learning (ML) can be used to improve the performance of IDS in 5G environments. Bouke and Abdullah have provided an in-depth empirical evaluation of ML-based IDS models and clearly focused on the problem of data leakage leading to the invalidity of evaluation outcomes [1]. Expanding on this line, future pieces of works have explored classical and learning-based IDS solutions that are specific to 5G traffic properties [2], and collaborative and federated IDS solutions, targeting distributed environments [3]. General surveys also point to the shift to more ML-driven and deep learning-driven IDS solutions, which have new abilities as well as constraints [4].

There have been a number of initiatives specifically directed at 5G network infrastructures, such as software-defined and virtualized environments to augment real-time intrusion detection and prevention services [5]. Additional research has discussed security as a network feature in 5G architectures with a focus on performance and scalability aspects [6]. Recent methods have considered domain adaptation and anomaly detection methods to adapt to traffic heterogeneity in 5G deployments [7]. Regardless of these developments, the majority of current IDS studies of 5G networks are focused on the detection accuracy based on a centralized or cloud-based assumption. This leads to the emerging requirement of leakage-free, edge-conscious designs of IDS to be able to run within realistic deployment conditions.

2 Motivation

The development of 5G networks has broadened the attack surface of communication infrastructure, especially advanced attacks on both the data and the control planes. Early detection of intrusions near the traffic source has thus also gained importance, particularly in alleviating signaling-based and high-rate attacks, which can quickly spread across network elements. Nevertheless, most of the currently available IDS solutions implicitly expect to have centralized processing and plenty of computational resources, which constrain their use at the network edge, where memory, computation, and latency factors prevail.

Meanwhile, with the introduction of TinyML and lightweight inference paradigms, it has been established that even with strict resource budgets, it is possible to make meaningful machine learning-based decisions. Such techniques as feature pruning and quantization have also demonstrated that it is possible to achieve significant efficiency improvements without making any fundamental changes to the detection logic. Simultaneously, network slicing in 5G means the incorporation of heterogeneous service demands, as the traffic nature and threat profiles can differ greatly across slices. This heterogeneity drives the requirement to have slice heterogeneity and not homogeneity in terms of IDS behavior.

With such advances, deployment-based evaluation has been underserved with inadequate focus on the combined evaluation of detection perception and edge plausibility in strict, leakage-free conditions. All these reasons prompted the creation of a lean, slice-conscious IDS that has trade-offs between detection stability and realistic edge limits, and which follows a high-discipline evaluation methodology.

3 Related Work

It is pointed out that leakage-free evaluation is a crucial need to provide a fair evaluation of machine learning-based IDS in 5G environments, especially with session-aware and chronological separation of data. Machine learning has also been extensively investigated to detect intrusion in 5G and beyond networks, and researchers have studied the appropriateness of algorithms in the context of different traffic conditions. The paradigms of distributed learning, including federated learning, have also been explored in order to solve the issue of privacy and scale in IDS implementations.

Surveys of classical and deep learning methods on network intrusion detection have been analyzed in detail, with their advantages and feasibility constraints described. Software-defined and virtualized 5G

infrastructures IDS designs have been focused on real-time detection needs, but tend to be based on complex designs. The 5G security issues have also been examined at the system level, where the need to incorporate security functions during the entire network service operations has been emphasized. It has been suggested to use domain adaptation methods to increase the strength of anomaly detection in changing traffic patterns in 5G networks.

Specific defense systems against control-plane threats have dealt with the problem of signaling-based denial-of-service attacks, and this clearly highlights the importance of early detection functionality [8]. IDS solutions that make use of deep learning have also been used in 5G-enabled IoT setups, but these solutions have computationally intensive requirements, which pose challenges to their deployment [9]. Pruning and quantization have been systematically assessed as an inference overhead reduction technique in model compression studies [10], and surveys into TinyML have summarized issues and opportunities of on-device intelligence [11].

The network slicing security has been discussed based on a network survey, which indicates slice-specific weaknesses and protection needs [12]. There have been first projects into slice-specific modeling of IDS that have shown promise but have not been deployed for validation [13]. Wider scopes of IDS surveys remain to display disjoints between experimentation and operational requirements [14].

Studies of the security issues in 5G have identified complexity in design and the necessity of coordinated protection solutions [15], and testbeds have shown that, in practice, implementing advanced security capabilities at the edge is challenging [16]. Theory Foundational works on deep learning developed the theoretical foundations of contemporary learning systems [17], and practical frameworks have supported the adoption of deep learning systems [18]. In sequence modelling and attention mechanisms, further development enabled learning to be applied to time-varying data [19], and convolutional architectures proved to be large-scale classification [20]. The initial versions of the backpropagation gave the foundations to such developments [21].

Resource-efficient learning is practicable in the conditions of very limited memory [22], and the dimensionality reduction by feature selection methods has long practiced [23]. Federated learning has been formalized as a paradigm of collaborative training of distributed systems [24], and IDS datasets surveys have revealed biases and evaluation problems [25]. Scalability in public networks has also been ventured into by autonomous and distributed IDS structures [26]. The need to address security, privacy, and trust issues in the IoT settings has already been extensively reported [27], as well as methodological recommendations on how to conduct a rigorous literature synthesis and assessment [28]. Comparative studies of centralized, on-device, and federated IDS designs have demonstrated trade-offs between deployability and performance [29], and the concept and application of federated learning have been discussed in fundamental terms [30]. Neural networks have been quantized to approach computational efficiency [31], and were simplified to the extreme with binary networks, which have shown extra resource savings [32]. Alternative efficiency perspectives (large-scale distributed learning systems [33]) and scalable tree-based models [34] are available, and extensive surveys put deep learning methods and applications into perspective [35]. Decades of research about IoT security have highlighted enduring limitations at the edge setting [36], and current surveys about network slicing security have condensed attack definitions and protection measures to apply to 5G deployments [37].

4 Research Gap

The current literature review indicates that there is notable advancement in the field of intrusion detection using ML on 5G networks, but there are still numerous gaps in the field. First, despite the explicit focus on leakage-free evaluation in recent literature [1], the majority of IDS research still focuses on detection accuracy, but does not consider deployment-related measurements like model size, computation complexity, or inference latency. Second, although much work exists on deep learning-based IDS [9], the scalability of these models to the network edge is still insufficient, which stimulates the investigation of classical ML with feature pruning and quantization [10, 22, 23].

Third, although network slicing security has received growing interest [12, 37], slice-aware IDS architectures have not been studied extensively, especially on realistic evaluation pipelines. The slice-specific methods that exist [13] are usually not leakage-free and do not measure efficiency-performance trade-offs

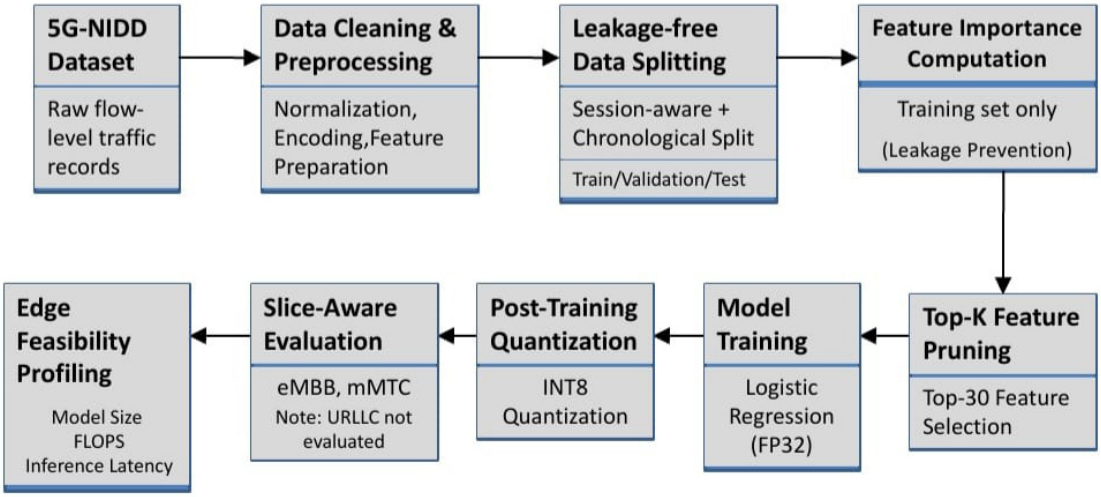

Figure 1. End-to-End workflow of the proposed Leakage-Free TinyML-Oriented edge IDS.

that pertain to edge deployment. Lastly, distributed and federated IDS systems [3, 26, 29] provide scalability at the cost of system complexity, which might not be appropriate to support a lightweight edge inference.

As a result, an IDS framework with collaborative focus on leakage-free evaluation, lightweight model design, and slice-aware behavior, and an explicit focus on quantifying edge feasibility by simulation-based measures, is clearly needed. This gap is the focus of the current work, which suggests and assesses an IDS based on TinyML and deployed on the edge of 5G network slices in a leakage-free environment.

5 Methodology

5.1 System Overview and Workflow

This work proposes a lightweight, edge-oriented intrusion detection framework for 5G networks that explicitly enforces leakage-free evaluation while remaining compatible with the constraints of TinyML. The overall system workflow is illustrated in Figure 1.

The pipeline begins with raw flow-level traffic records extracted from the 5G-NIDD dataset. After data cleaning and preprocessing, a leakage-free split is enforced using a session-aware and chronological separation strategy to ensure that no temporal or session overlap occurs across training, validation, and test sets. Feature importance is then computed exclusively on the training subset and used to perform Top-K feature pruning. A classical Logistic Regression model is trained on the pruned feature space and subsequently subjected to post-training INT8 quantization. Finally, the quantized model is evaluated in a slice-aware manner, and its edge feasibility is assessed using software-based profiling metrics, including model size, FLOPs, and inference latency.

This workflow ensures that detection performance and deployment feasibility are evaluated jointly under realistic constraints.

5.2 Input–Output Data Flow

The detailed input–output transformation performed by the proposed IDS is depicted in Figure 2.

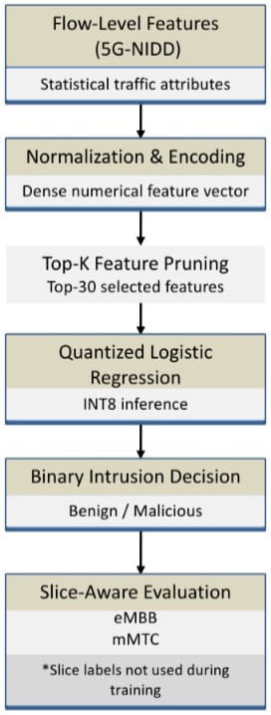

Figure 2. Input-Output dataflow of the proposed TinyML-Oriented edge IDS.

The input to the system consists of flow-level statistical

features derived from 5G-NIDD traffic records. These features are first normalized and encoded to form a dense numerical feature vector. Feature pruning is applied to this vector to retain only the Top-K most informative features, reducing dimensionality and computational complexity. The pruned feature vector is then processed by a quantized Logistic Regression classifier, producing a binary intrusion decision. Slice labels (eMBB, mMTC) are not used during training but are employed during evaluation to analyze slice-specific inference behavior.

Pruning and quantization are explicitly highlighted in the data flow, emphasizing their role in achieving edge feasibility without altering the underlying detection logic.

The proposed model employs logistic regression as the core classifier due to its simplicity, interpretability, and suitability for resource-constrained edge deployment. Let $x \in \mathbb{R}^d$ denote the normalized feature vector extracted from 5G network flows, where d is the dimensionality of the original feature space. The model computes the probability of a flow being malicious as:

$$\hat{y} = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}}, \tag{1}$$

where $w \in \mathbb{R}^d$ is the weight vector, b is the bias term, and $\sigma(\cdot)$ denotes the sigmoid activation function. The model is trained by minimizing the binary cross-entropy loss over N training samples:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]. \tag{2}$$

To reduce model complexity and memory footprint, we perform feature pruning based on the magnitude of learned weights. The importance score for feature j is defined as $I_j = |w_j|$, and the Top-K feature set is selected as $S = \arg\max_K I_j$. After retraining on the reduced feature space $x^{(K)} \in \mathbb{R}^K$, we apply INT8 post-training quantization to the weights:

$$w^{(q)} = \text{round}\left(\frac{w}{\alpha}\right), \quad \alpha = \frac{\max(|w|)}{127}, \tag{3}$$

where α is the scaling factor derived from the dynamic range of w. Table 1 summarizes the key symbols and variables used throughout the formulation and algorithm.

Table 1. Notation and description.

Symbol	Description
D	Full dataset (5G-NIDD flow-level records with features, labels, session identifiers, timestamps, and slice tags)
x	Normalized input feature vector (full feature space)
d	Number of original features before pruning
y	Ground-truth binary class label (benign/malicious)
$\hat{y}$	Predicted output label from the classifier
w	Logistic Regression weight vector (learned parameters)
b	Logistic Regression bias term
I	Feature importance scores computed from the trained model
K	Top-K pruning threshold (number of retained features)
S	Set (or index list) of Top-K selected features

5.3 Leakage-Free Evaluation Protocol

A critical requirement for reliable intrusion detection in 5G networks is ensuring that the evaluation protocol does not introduce temporal data leakage, which would artificially inflate performance metrics. Traditional random splitting of samples fails to account for the temporal correlation between network flows belonging to the same session. To address this, we design a session-aware, temporally sequential split that guarantees no data leakage between training, validation, and test partitions.

Algorithm 1 implements this leakage-free protocol alongside the model training, pruning, quantization, and slice-wise evaluation pipeline. The algorithm takes the full dataset D and pruning threshold K as inputs and outputs a quantized logistic regression model suitable for TinyML deployment, along with per-slice detection performance metrics.

5.4 Experimental Setup and Configuration

The 5G-NIDD dataset was used in experiments to test whether features of flow-level statistical statistics with labeled intrusion events are applicable. Traffic labels were categorized as benign and malicious. A tight leakage-free evaluation protocol was imposed by session-aware separation and then chronological order, where training, validation, and test sets did not have common session identifiers. This bifurcation approach

Algorithm 1: Leakage-Free TinyML-Oriented Edge IDS

Input: Dataset D, pruning threshold K
Output: Quantized LR model and slice-wise results

1. Load and clean D (duplicates, $\pm\infty$, missing values);
2. Convert labels to binary (Benign vs Malicious);
3. Group samples by session identifiers;
4. Sort sessions chronologically by timestamps;
5. Split sessions into train/val/test with no overlap;
6. Verify no session appears in >1 split;
if *overlap* **then**
  reassign
end
7. Select feature columns (exclude metadata);
8. Fit encoders/scaler on training only; transform all partitions to x;
9. Train LR (FP32) on training set using x;
10. Compute feature importance $I = |w|$, select Top-K set S;
11. Reduce all partitions to $x^{(K)}$ using S;
12. Retrain LR (FP32) on training set using $x^{(K)}$;
13. Compute scaling factor α from weight range;
14. Apply INT8 quantization: $w^{(q)} = \text{round}(w/\alpha)$;
15. Test inference on pruned test set to get $\hat{y}$;
16. Compute Accuracy, Precision, Recall, F1, confusion matrix;
17. **for** *each slice in* {*eMBB, mMTC*} **do**
  Filter test samples, run inference, compute slice-wise metrics;
end
18. **if** *URLLC absent* **then**
  report "URLLC not evaluated"
end
19. Profile model size, FLOPs, latency;
return outputs;

was aimed at avoiding both temporal and contextual leakage between levels of evaluation.

It was trained on an intrusion detection model on a Logistic Regression classifier with L2 regularization and a standard solver that is compatible with the implementation, and no deep learning architecture was utilized. The model coefficients learned in the model were used as feature importance, and Top-K pruning was used, only keeping K = 30 features, which were chosen empirically between validation performance and efficiency factors. The trained model weights were then quantized using post-training INT8 quantization by software-based simulation, without making the physical deployment assumption. Edge feasibility was evaluated by calculating the model size with data being serially read, estimating the FLOPs analytically using the feature dimensionality, and inference latency, which was calculated by profiling on the CPU. Accuracy, Precision, Recall, and F1-score were used to detect performance, and model size, FLOPs, and inference latency were used to determine deployment feasibility. Each methodological step was followed as set out to guarantee reproducibility and consistency among the codebase, methodology, and reported results.

6 Results and Discussion

6.1 Experimental Results Overview

The experimental analysis evaluated the effectiveness of detection and the feasibility of deployment of a TinyML-based intrusion detection system in a rigid, leakage-free protocol. Two major combinations of the model were described: (i) a full-feature Logistic Regression model, trained with 32-bit floating-point precision (LR-FP32), and (ii) a pruned, post-training INT8 quantized Logistic Regression model, only keeping the Top-30 most important features (LR-INT8). The models with intermediate pruning-only variants were used in model selection, and they are not addressed in the final comparative analysis.

All the reported findings were acquired with session-aware and chronological train-validation test splits to avoid data leakage. The analysis of performance was conducted on three complementary dimensions: overall detection capability, edge feasibility metrics, and slice-aware behavior of eMBB and mMTC traffic. This was done without any physical hardware that was installed and through software profiling.

6.2 Detection Performance Analysis

Table 2 provides a summary of the test-set detection performance of the full-feature FP32 Logistic Regression model and the pruned INT8 Logistic Regression model in the leakage-free constraint. The pruned INT8 model maintained and slightly improved the detection performance of the full feature FP32 when it comes to accuracy and F1-score. In particular, there was a constant performance in detection with significant decreases in feature dimensionality and numerical precision. All model variants were similar in overall classification balance, which was assessed

Table 2. Detection performance comparison across models.

	Model	Accuracy	F1	Model Size (KB)	FLOPs	Latency(ms)
0	LR-FP32(Full)	0.928	0.928	0.335938	170	0.248065
1	LR-INT8(Top-30)	0.936	0.937	0.030273	60	0.178139

through stable values of accuracy and F1-score values. The findings show that feature pruning and post-training quantization did not affect the detection performance of the proposed leakage-free assessment protocol.

6.3 Impact of Feature Importance–Based Pruning

The pruning based on feature importance showed a quantifiable effect on the compactness of the model and or computational efficiency, but did not affect the detection performance. The decreasing dimension of features of the complete feature set to the Top-30 ones led to slight changes in the detection measures, meaning that only a few discriminative features described most of the predictive power.

The chosen Top-30 threshold was a compromise between the size and performance measured in the validation stage, and allows for gaining significant efficiency without causing instability in the classification results. These results demonstrate that classical linear models are appropriate to reduce the features of aggressions in edge-based intrusion detection.

6.4 Effect of INT8 Quantization

INT8 quantization also helped to minimize the computation cost of the pruned model after training. The quantized model was found to have substantial size and complexity reduction, finding relative to its FP32 counterpart without altering detection performance. Quantization did not add any form of observable degradation to accuracy or F1-score.

Quantization was implemented after training the models and testing by simulation with software, and it was verified that the behavior of detection was consistent with the trained FP32 model. These findings indicate that post-training quantization is a valid tactic that can be employed to improve edge feasibility without the need to retrain or alter the architecture.

6.5 Edge Feasibility Evaluation

The criteria of edge feasibility were model size, floating-point operations (FLOPs), and inference latency, which were determined through CPU-based profiling. The pruned INT8 model showed a significant downsizing of the model compared to the full-feature FP32 baseline, with an order-of-magnitude reduction in the memory footprint, as shown in Figure 3. FLOPs were also minimized, which indicates a lesser computational requirement in the inference.

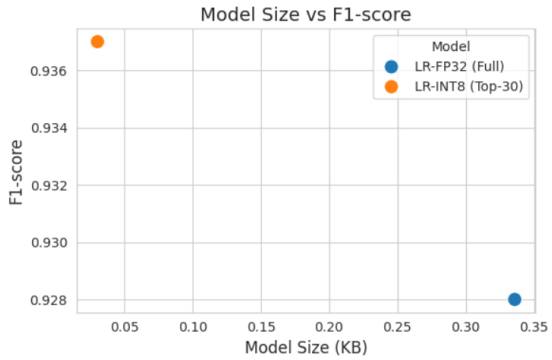

Figure 3. Model Size vs. F1-Score Trade-off. Relationship between model size and F1-score for full-feature and pruned INT8 Logistic Regression models.

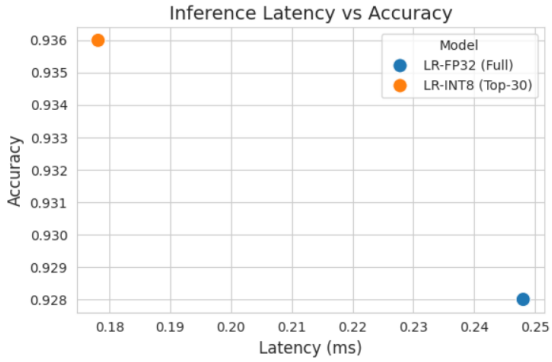

Figure 4. Inference Latency vs. Accuracy Trade-off. Comparison of inference latency and accuracy for the evaluated Logistic Regression model variants under CPU-based simulation.

Measurements of inference latency showed that both the pruned and quantized model consistently reduced their execution time, and execution time of less than a millisecond was realized in the simulated environment. The trade-off plots between model size and F1-score (illustrated in Figure 3) as well as inference latency and accuracy (illustrated in Figure 4) indicated efficiency improvement without negative impact on detection performance, as shown in Figure 5. There were

no statements on real-time guarantees or hardware implementation.

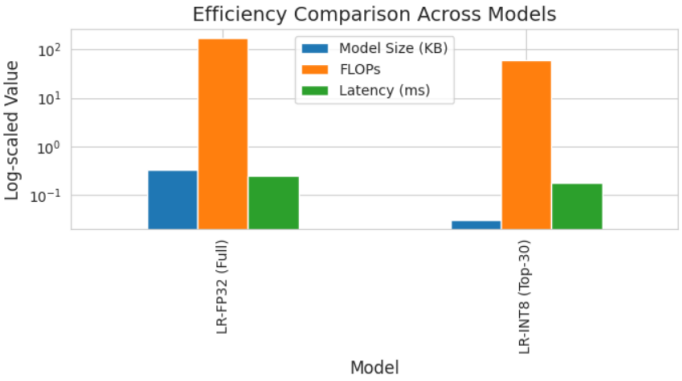

Figure 5. Efficiency Comparison Across Models. Log-scale comparison of model size, FLOPs, and inference latency for full-feature FP32 and pruned INT8 Logistic Regression models.

The joint comparison of detection performance (Accuracy and F1-score) and efficiency-related measures (model size and inference latency) of the assessed Logistic Regression configurations is shown in Figure 6. The radar plot allows the representation of the effect of feature pruning and INT8 quantization on efficiency gains and similar detection performance to the full-feature FP32 model.

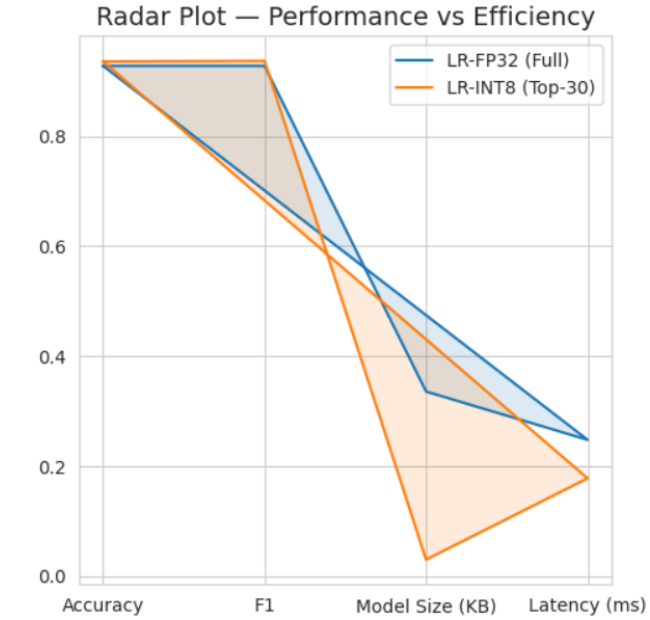

Figure 6. Radar Plot-Performance vs. Efficiency. Radar plot illustrating the trade-off between detection performance and efficiency metrics across evaluated model variants.

6.6 Slice-Aware Performance Analysis

Slice-aware evaluation was conducted using slice-specific confusion matrices for enhanced Mobile Broadband (eMBB) and massive Machine-Type Communications (mMTC) traffic.

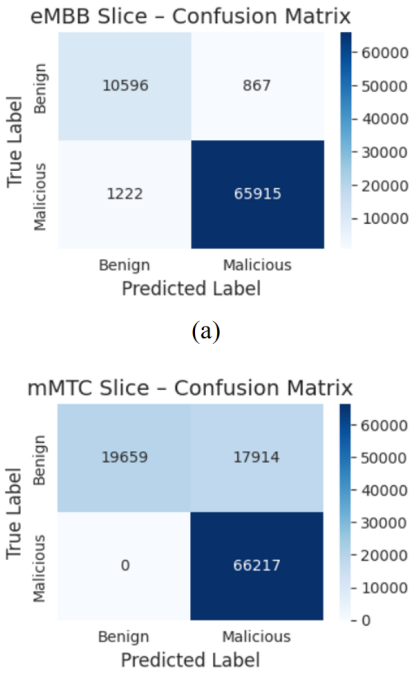

Figure 7. Radar Plot-Performance vs. Efficiency. Radar plot illustrating the trade-off between detection performance and efficiency metrics across evaluated model variants.

According to Figure 7(a), the eMBB slice also demonstrated steady classification, which is indicated by a concentration of true benign and true malicious samples on the primary diagonal of the confusion matrix. This distribution revealed that there was stable segregation between benign and malicious flows in the eMBB slice under the leakage-free appraisal protocol. In Figure 7(b), the mMTC slice confusion matrix exhibited relatively more benign traffic misclassifications, indicating that there is more variability in the traffic attributes that are correlated with the large-scale machine-type communications. In spite of this variation, malicious traffic identification in the mMTC slice was otherwise uniform, with no false negative effects of malicious samples. The test set that was evaluated did not include any samples of the nature of Ultra-Reliable Low-Latency Communications (URLLC) slice; thus, no conclusions were made about the URLLC behavior.

6.7 Discussion Summary and Practical Implications

The conducted experiment proved that the TinyML-oriented IDS relying on classical Logistic Regression was able to attain stable results in detection when evaluated in leakage-free conditions and

significantly enhanced the feasibility of deployment. The pruning of features and INT8 quantization were used together to achieve a smaller model size, less cost in terms of computations, and shorter latency in inferences without affecting the reliability of the classification.

Slice-aware analysis also revealed that the offered approach was active with the considered 5G slices (eMBB and mMTC), which indicates the need to take service-specific behavior into account during the IDS design. In general, the results demonstrate the viability of lightweight, deployment-focused IDS solutions to the 5G edge environment in case stringent evaluation discipline and efficiency-conscious design are employed in unison.

7 Comparative Analysis

Bouke et al. [1] developed stringent research that provided a base of leakage-free analysis of machine learning-based intrusion detection systems in 5G networks through imposing session-conscious and chronological data partitioning. Nevertheless, they concentrated more on the detection performance in the centralized deployment assumptions and did not specifically discuss the feasibility of deployment in the network edge. The current paper, on the contrary, brought leakage-free IDS assessment to an edge-based view with efficiency-conscious design and analysis. Although a classical Logistic Regression model is adopted within the same leakage-free discipline, feature importance-based pruning and post-training INT8 quantization were introduced in the study to aim at minimizing model complexity. In addition to measures of detection, model size, computational complexity, and inference latency were measured using software-based profiling, thus making the evaluation consistent with the edge constraints in practice. Also, slice-awareness behavior of eMBB and mMTC traffic was studied, which includes the service heterogeneity of the 5G networks. Instead of focusing on improving the detection scores, this contribution showed that leakage-free IDS models could be tailored to resource-constrained 5G edge environments systematically and can be used to preserve the same detection behavior.

8 Conclusion and Future Scope

This paper offered a TinyML-based intrusion detection framework of 5G networks that integrated both rigorous leakage-free analysis and deployment-aware design. The research maintained a high level of methodology due to the use of session-aware and chronological data splitting, as well as evaluating a lightweight IDS using classical Logistic Regression. Top-30 pruning based on feature importance, and standardized quantization to INT8 after training was used to simplify the models, and their effects were assessed based on simulation-based metrics of edge feasibility, such as model size, computational complexity, and inference latency. The findings revealed that the detection behavior did not diminish in such efficiency limits and thus established that the high levels of reduction in resource requirements could be made without affecting the reliability of classification. Slice-conscious testing also reflected that the functionality of eMBB and mMTC traffic was the same, and thus the significance of service heterogeneity in the consideration of 5G security analysis.

This assessment, in the future, can be expanded to other 5G-related datasets so that the generalization in various traffic conditions can be evaluated more exhaustively. With appropriate data made available, the framework can likewise be investigated on the underexplored slices, like URLLC, to have a broader slice-level view. Lastly, although this study used a software-based profiling, this can be improved in the future by incorporating hardware-in-the-loop experimentation to demonstrate the efficiency trends in the observed conditions when executed in the real-world on the edges.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Bouke, M. A., & Abdullah, A. (2024). An empirical assessment of ML models for 5G network intrusion detection: A data leakage-free approach. *e-Prime-Advances in Electrical Engineering, Electronics and Energy, 8*, 100590. [CrossRef]

[2] Imanbayev, A., Tynymbayev, S., Odarchenko, R., Gnatyuk, S., Berdibayev, R., Baikenov, A., & Kaniyeva, N. (2022). Research of machine learning algorithms for the development of intrusion detection systems in 5G mobile networks and beyond. *Sensors, 22*(24), 9957. [CrossRef]

[3] Buyuktanir, B., Altinkaya, Ş., Karatas Baydogmus, G., & Yildiz, K. (2025). Federated learning in intrusion detection: advancements, applications, and future directions. *Cluster Computing, 28*(7), 473. [CrossRef]

[4] Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32*(1), e4150. [CrossRef]

[5] Bocu, R., & Iavich, M. (2022). Real-time intrusion detection and prevention system for 5G and beyond software-defined networks. *Symmetry, 15*(1), 110. [CrossRef]

[6] Malik, S., & Bera, S. (2024, July). Security-as-a-function in 5g network: Implementation and performance evaluation. In *2024 International Conference on Signal Processing and Communications (SPCOM)* (pp. 1-5). IEEE. [CrossRef]

[7] Kim, H. J., Lee, J., Park, C., & Park, J. G. (2022, October). Network anomaly detection based on domain adaptation for 5g network security. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 976-980). IEEE. [CrossRef]

[8] Silva, R. S., Meixner, C. C., Guimaraes, R. S., Diallo, T., Garcia, B. O., de Moraes, L. F., & Martinello, M. (2020). REPEL: A strategic approach for defending 5G control plane from DDoS signalling attacks. *IEEE Transactions on Network and Service Management, 18*(3), 3231-3243. [CrossRef]

[9] Yadav, N., Pande, S., Khamparia, A., & Gupta, D. (2022). Intrusion detection system on IoT with 5G network using deep learning. *Wireless Communications and Mobile Computing, 2022*(1), 9304689. [CrossRef]

[10] Shabir, M. Y., Torta, G., & Damiani, F. (2025). TinyML model compression: A comparative study of pruning and quantization on selected standard and custom neural networks. *Telecommunication Systems, 88*(4), 1-21. [CrossRef]

[11] Heydari, S., & Mahmoud, Q. H. (2025). Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions. *Sensors, 25*(10), 3191. [CrossRef]

[12] Dangi, R., Jadhav, A., Choudhary, G., Dragoni, N., Mishra, M. K., & Lalwani, P. (2022). Ml-based 5g network slicing security: A comprehensive survey. *Future Internet, 14*(4), 116. [CrossRef]

[13] Akpan, V. A., Njoku, E. C., & Obi, E. I. (2025). Slice-specific machine learning models for intrusion detection in 5G telecommunication networks. *International Journal of Wireless Communications and Mobile Computing, 12*(2), 93-118. [CrossRef]

[14] Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discover Artificial Intelligence, 5*(1), 314. [CrossRef]

[15] Dutta, A., & Hammad, E. (2020, September). 5G security challenges and opportunities: A system approach. In *2020 IEEE 3rd 5G world forum (5GWF)* (pp. 109-114). IEEE. [CrossRef]

[16] Pepito, R., & Dutta, A. (2021). Open source 5G security testbed for edge computing. In *2021 IEEE 4th 5G World Forum (5GWF)* (pp. 388-393). IEEE. [CrossRef]

[17] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, 102419. [CrossRef]

[18] Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University-Computer and Information Sciences, 34*(4), 1595-1623. [CrossRef]

[19] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 1-15. [CrossRef]

[20] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective, 25*(1-3), 18-31. [CrossRef]

[21] Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2019). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications, 12*(2), 493-501. [CrossRef]

[22] Kumar, A., Goyal, S., & Varma, M. (2017, July). Resource-efficient machine learning in 2 kb ram for the internet of things. In *International conference on machine learning* (pp. 1935-1944). PMLR.

[23] Bolon-Canedo, V., Sanchez-Marono, N., & Alonso-Betanzos, A. (2011). Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. *Expert Systems with Applications, 38*(5), 5947-5957. [CrossRef]

[24] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future

directions. *IEEE signal processing magazine, 37*(3), 50-60. [CrossRef]

[25] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & security, 86*, 147-167. [CrossRef]

[26] Mahmoodi, A. B. Z., Sheikhi, S., Peltonen, E., & Kostakos, P. (2023). Autonomous federated learning for distributed intrusion detection systems in public networks. *IEEE access, 11*, 121325-121339. [CrossRef]

[27] Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems, 78*, 544-546. [CrossRef]

[28] Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1-10). [CrossRef]

[29] Rahman, S. A., Tout, H., Talhi, C., & Mourad, A. (2020). Internet of Things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network, 34*(6), 310-317. [CrossRef]

[30] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST), 10*(2), 1-19. [CrossRef]

[31] Jiang, C. (2021, November). Efficient quantization techniques for deep neural networks. In *2021 International conference on signal processing and machine learning (CONF-SPML)* (pp. 271-277). IEEE. [CrossRef]

[32] Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016, September). Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision* (pp. 525-542). Cham: Springer International Publishing. [CrossRef]

[33] Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 22*(2), 869-904. [CrossRef]

[34] Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE communications surveys & tutorials, 22*(3), 1646-1685. [CrossRef]

[35] Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials, 21*(3), 2224-2287. [CrossRef]

[36] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146-164. [CrossRef]

[37] De Alwis, C., Porambage, P., Dev, K., Gadekallu, T. R., & Liyanage, M. (2024). A survey on network slicing security: Attacks, challenges, solutions and research directions. *IEEE Communications Surveys & Tutorials, 26*(1), 534-570. [CrossRef]

Phalguni Patnaik is an undergraduate student pursuing a B.Tech. in Computer Science Engineering at NIST University, with a strong interest in Data Analytics, Artificial Intelligence, and Machine Learning. She focuses on building practical and research-driven solutions and has worked on projects involving predictive modeling and AI applications, demonstrating a structured approach to problem formulation, experimentation, and evaluation. (Email: phalguni.patnaik.cse.2022@nist.edu)

Susrita Mishra is currently pursuing a B.Tech. in Computer Science Engineering at NIST University and has a strong academic inclination toward Data Analytics, Artificial Intelligence, and Machine Learning. Her interests lie in translating theoretical concepts into applied and research-aligned solutions, with hands-on experience in AI-based modeling and analytical systems. (Email: susrita.mishra.cse.2022@nist.edu)

Bandhan Panda serves as an Assistant Professor in the Department of Computer Science and Engineering at NIST University, Berhampur. With over two years of academic experience in the fields of engineering and computer applications, he has actively guided and mentored students in their educational and professional development. He is presently pursuing a Ph.D. in Computer Science at Berhampur University, focusing on emerging research areas within the discipline. His dedication to teaching, research, and innovation underscores his commitment to academic excellence and his passion for fostering student success. (Email: bandhan.panda@nist.edu)

Santosh Kumar Kar is a Senior Assistant Professor in the Department of Computer Science and Engineering at NIST University, Berhampur. He earned his Ph.D. in Computer Science and Engineering in 2025. With over 17 years of teaching experience in the engineering and Application domain and two years of industry exposure, he combines academic rigor with practical insight. His primary areas of interest include Software Engineering and Artificial Intelligence, where he actively contributes through teaching, research, and mentorship, fostering innovation and excellence among students and peers. (Email: email@email.com)