Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084409

ARTICLE

Quantized Intrusion Detection for Resource-Constrained IoT: A Comparative Evaluation of Efficiency and Adversarial Robustness

Saeed Ullah[1], Junsheng Wu[1,*], Mian Muhammad Kamal[2,*], Mohammed K. Alzaylaee[3] and Heba G. Mohamed[4,5]

[1]School of Software, Northwestern Polytechnical University, Xi'an, 710072, China
[2]School of Electronic and Communication Engineering Quanzhou University of Information Engineering, Quanzhou, 362000, China
[3]Department of Computing, College of Engineering and Computing in Al-Qunfudhah, Umm AL-Qura University, Makkah, Saudi Arabia
[4]Department of Electrical Engineering, College of Engineering, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[5]Electrical Department, College of Engineering, Alexandria Higher Institute of Engineering and Technology, Alexandria, 21421, Egypt

*Corresponding Authors: Junsheng Wu. Email: wujunsheng@nwpu.edu.cn; Mian Muhammad Kamal. Email: mianmuhammadkamal@qzuie.edu.cn

Received: 22 April 2026; Accepted: 04 June 2026; Published: 23 July 2026

ABSTRACT: The proliferation of Internet of Things (IoT) devices has introduced unprecedented security challenges, necessitating efficient intrusion detection systems (IDS) capable of operating under severe resource constraints. This research presents a hardware-informed empirical study of quantized neural-network-based intrusion detection for resource-constrained IoT platforms, using an ARM Cortex-M4 deployment target as a reference. We evaluate FP32, FP16, and INT8 TensorFlow Lite model variants derived from a lightweight 1D-CNN and assess their trade-offs in clean-data accuracy, model size, estimated inference latency, estimated energy consumption, and adversarial robustness. INT8-quantized model achieves 99.10% accuracy on clean data while maintaining 97.50% adversarial accuracy under Projected Gradient Descent (PGD) attacks with perturbation budget = 0.3. The quantized model achieves 12.0× latency reduction (0.083 vs. 0.995 ms) and 92.7% energy reduction (0.0083 vs. 0.1135 mJ) when compared to FP32. The memory footprint of the model is reduced by 55.7% from 58.02 to 25.72 KB. Our comprehensive analysis includes confusion matrices, ROC curves (AUC = 0.9964 for INT8), adversarial robustness heatmaps, and statistical significance testing via McNemar's test. The results establish INT8 quantization as a viable solution for deploying robust IDS on resource-constrained IoT devices, achieving practical deployment feasibility without reducing detection performance.

KEYWORDS: IoT security; intrusion detection; neural network quantization; adversarial robustness; energy-efficient computing; TensorFlow lite

1 Introduction

IoT technology has shifted quickly from an emerging idea to something embedded in global infrastructure, tying together millions of devices in cities, factories, and healthcare settings [1]. However, the proliferation of IoT has precipitated significant security challenges, large DDoS attacks being among the most visible, along with the ongoing struggle to protect devices that have very few resources [2]. Global numbers are expected to jump from 19.8 billion devices in 2025 to over 40.6 billion by 2034 [3]. What this means in practice is a far bigger attack surface, since many of these billions of devices sit in unfriendly

environments with constrained power and no real ability to run heavy security software [4–6]. Consequently, these devices constitute high-value targets for adversarial exploitation [7]. A botnet-powered DDoS can take down digital services from anywhere in the world [8,9]. In response, modern intrusion detection has moved toward machine learning and deep learning, stepping away from older signature-based approaches. These models learn what normal and abnormal traffic looks like, catching zero-day threats that signatures would miss [10]. This creates a fundamental tension between security requirements and the operational constraints of edge hardware. Running a Convolutional Neural Network consumes resources that small devices simply cannot provide [11,12]. Traditional IDS setups do not transfer well to edge hardware with 64–512 KB of RAM, MCUs running at 80–200 MHz, and batteries that need to last for years [13]. High-accuracy models assume computing headroom that chips like the ESP32 or ARM Cortex-M cannot offer [14]. This tension has motivated research into TinyML and model compression, with quantization getting a lot of attention. Quantization cuts parameter precision, shifting from 32-bit floating-point to 8-bit integers, bringing latency and memory use down. The impact of quantization on adversarial robustness remains insufficiently characterized. Computer vision research already shows that neural networks can be fooled by tiny, nearly invisible changes to an input [15]. The same idea applies to network packets; an attacker could craft traffic that slides straight past an ML-based IDS. Recent lightweight IoT IDS work tends to study efficiency and security in isolation [16]. QDNN-IDS and SMEESI TinyML [17] show that quantized networks bring down energy use and memory on edge devices, but neither digs into whether lower precision opens new adversarial doors. TinyML security studies like [18] look at adversarial transferability but stop short of measuring real energy costs on specific hardware. Another blind spot is gradient masking in non-differentiable INT8 models, which can make a model look more robust than it actually is. This work addresses these gaps together, measuring hardware realities on (targeting ARM Cortex-M4 [19]) in simulation while testing adversarial robustness through transfer attacks that avoid gradient masking.

1.1 Motivation and Problem Statement

The deployment of neural network-based intrusion detection systems on IoT platforms presents three fundamental challenges. First, inference latency: hundreds of milliseconds of latency introduced by running models (neural networks) on resource-constrained devices, makes them far too slow for real-time intrusion detection. Second, energy sustainability: sensors designed to run for months or years quickly deplete their batteries when they constantly process packets. Third, adversarial vulnerability: subtle input perturbations can induce misclassification, and quantization may amplify this susceptibility through gradient masking, a phenomenon that conceals true model weaknesses beneath apparent robustness [20]. This work posits that efficiency and security constitute a unified optimization problem rather than independent objectives. Empirical evaluation of quantized networks demonstrates that quantization is not only about reducing the model size, but can also be used as a defense against attacks.

1.2 Contributions

This paper makes the following contributions.

- **Comprehensive Quantization Analysis.** A thorough quantization analysis of FP32, FP16, and INT8 on the CICIoT2023 dataset across accuracy, model size, latency, and energy consumption.
- **Adversarial Robustness under Quantization.** Models Robustness under PGD attacks for different perturbation budgets (ε = 0.05, 0.1, 0.2, 0.3) and perform transfer attacks for INT8 to avoid gradient masking.
- **Statistical Validation.** Confusion matrices, ROC curves, precision-recall curves, the specificity, False Positive Rate (FPR), and the McNemar test for a thorough comparison of the models.

- **Practical Deployment Guidelines.** Latency and energy analysis indicate that INT8 is a better deployment option than FP32, and TensorFlow Lite implementation is provided.

The remainder of this paper is organized as follows: Section 2 reviews related work in IoT security and quantization. Section 3 details our methodology, Section 4 presents the experimental results on accuracy, robustness, and energy efficiency, Section 5 present the Discussion, followed by the conclusion in Section 6.

2 Related Works

With the increasing proliferation of IoT and the launch of 5G, the traditional signature-based Intrusion Detection System (IDS) needs an upgrade to a more sophisticated and reliable form of IDS. This advanced IDS uses deep learning techniques. There has been a division in current research, which has focused on two areas (1) improving the robustness of anomaly detection systems to sophisticated adversarial attacks and (2) developing techniques to make the learning process more efficient in situations where there are hardware restrictions.

2.1 Adversarial Robustness and Advanced Detection in IoT

DL-based IDSs are becoming common and are facing adversarial attacks. There are a few studies that have examined vulnerability to rule-based adversarial attacks, finding that classifiers can be fooled by minor adjustments to feature inputs [21]. To improve robustness, adversarial training has been developed, though its iterative retraining demands substantial computation [22,23]. Yuan et al. in [24] proposed a method employing Local Intrinsic Dimensionality (LID) and Label Spreading to identify adversarial points in high-dimensional space. However, calculating LID scores requires complex manifold estimation, which remains computationally expensive for real-time use. Developing zero-day attack detection, the distributed ADHDN framework [25] uses a GAN-based honeynet to detect both threats and novel attacks. In the same way, DeepTransIDS [26] uses a self-attention mechanism for 5G Non-IP Data Delivery to capture the long-range dependencies in traffic. While these approaches have achieved state of the art detection rates, the complex architectures (GANs and Transformers) usually require more computing power than is available in low power IoT devices [27,28]. For a high detection rate by hybrid balancing in [29] used SVWS oversampling to reach 99.73% accuracy using MLP with probability-fed categorical boosting. However, none of these approaches addressed the issue of compression, adversarial robustness or embedded implementation.

2.2 Lightweight Architectures, Neural Network Quantization and Energy Optimization

In addition to robustness, a significant amount of work is being done to reduce the computational load of IDS for deployment on the edges of the network [30]. In [31], authors proposed a CNN-LSTM architecture with quantization for resource-constrained devices. A lighter architecture for IoT was obtained by reducing weight precision [32]. However, the quantized model was not tested against adversarial examples from [18]. A game-theoretic model in [33] optimized IDS power consumption by modeling defender-attacker interaction. While this prioritizes battery life, it does not adapt to changes in adversarial inputs encountered by the detection model. A suitable accuracy-cost compromise was achieved through optimization techniques [34]. The Firefly Algorithm-based CNN Hyperparameter Tuning (EFACNN) [35] minimized manual tuning requirements while achieving high accuracy. The traffic flow topological dependencies were well preserved by a Graph Neural Network model (FTG-Net-E) [36]. In Ref. [37] proposed ML-VIDS achieving 97.8% accuracy with sub-30W energy, but omitted adversarial robustness and quantization. In Ref. [38] designed a 15,913-parameter attention-based autoencoder for vehicular intrusion detection, yet also left out adversarial robustness and quantized implementation. These methods, while efficient, do not quantify energy cost or adversarial attack impact.

2.3 The Energy-Robustness Gap

Examination of existing research reveals a pronounced tension between robustness and energy efficiency. Most security work either reduces power consumption at the expense of adversarial robustness, or pursues energy efficiency while sacrificing security [39,40]. Examples of the former are the Lid computations described in [24] and the transformers discussed in [26]. The latter includes quantization, as described in [27,31] and the game-theoretic methods for saving energy discussed in [33]. Xu and Fang [41], achieved a 37% reduction in vehicular IDS latency through feature selection; however, they did not evaluate energy per inference or adversarial robustness, leaving the energy–robustness trade-off unexplored. There remains a lack of research that unifies these conflicting constraints. Currently, none of the existing frameworks effectively assess how much lightweight compression techniques [31], impair the adversarial defense mechanisms [21]. Moreover, no unified approach is in place to strike a balance where an IDS is both robust to interference and energy efficient enough for use at the network edge over a long period of time. This paper addresses this gap by proposing Lightweight Energy-Robust Detection possibility.

To explicitly position our contribution against the most closely related prior studies, we highlight the following distinctions in Table 1. In Ref. [31] did not explore robustness and energy in his work. In Ref. [24] approach for manifold estimation is still too expensive for a Cortex-M4. In Ref. [33] employed game theory for robust training however did not perform any quantization or adversarial tests. In Ref. [26] proposed an architecture that has highly computation power. In Ref. [21,30] did not analyze how quantization affects the behavior of the model. Our work is the first to combine (i) INT8 quantization with TFLite, (ii) PGD/transfer-based adversarial evaluation covering gradient masking, and (iii) Energy-latency profiling for resource-constrained IoT intrusion detection.

Table 1: Comparative study of literature.

Study	Year	Quantization	Latency	Energy	Adv. Robustness	Key Limitation
[21]	2021	✗ None	Not reported	Not reported	52.7%	Rule-based attacks only
[31]	2024	Yes (post-training + Pruning)	Not reported	Not reported	✗ None	No adversarial evaluation
[24]	2024	✗ None	Not reported	Not reported	60–90%	Incompatible with edge deployment
[30]	2024	✗ None	~<20 ms	<10% CPU	✗ None	No quantization possible
[40]	2024	Yes (8-bit/16-bit)	45 ms	28% reduction	92.5%	Limited implementation details
[33]	2024	✗ None	N/A	116 kJ	✗ None	Simulation-heavy
[26]	2025	✗ None	~2.78 ms	Not reported	✗ None	high compute cost
This work	2026	INT8 TFLite	0.083 ms	0.0083 mJ	97.5%	—

3 Proposed Methodology

To resolve the energy-resilience trade-off described in Section 2, a unified evaluation procedure is developed to quantify the compromises between system robustness against malicious activity and energy consumption. Our methodology consists of Seven distinct stages: (1) Dataset and Preprocessing, (2) Baseline Model Architecture (3) Training Configuration (4) Post-Training Quantization (5) Hardware-Informed Analytical Estimation (6) Adversarial Robustness Evaluation and (7) Statistical Validation and Extended Metrics. The overall workflow is illustrated in Fig. 1.

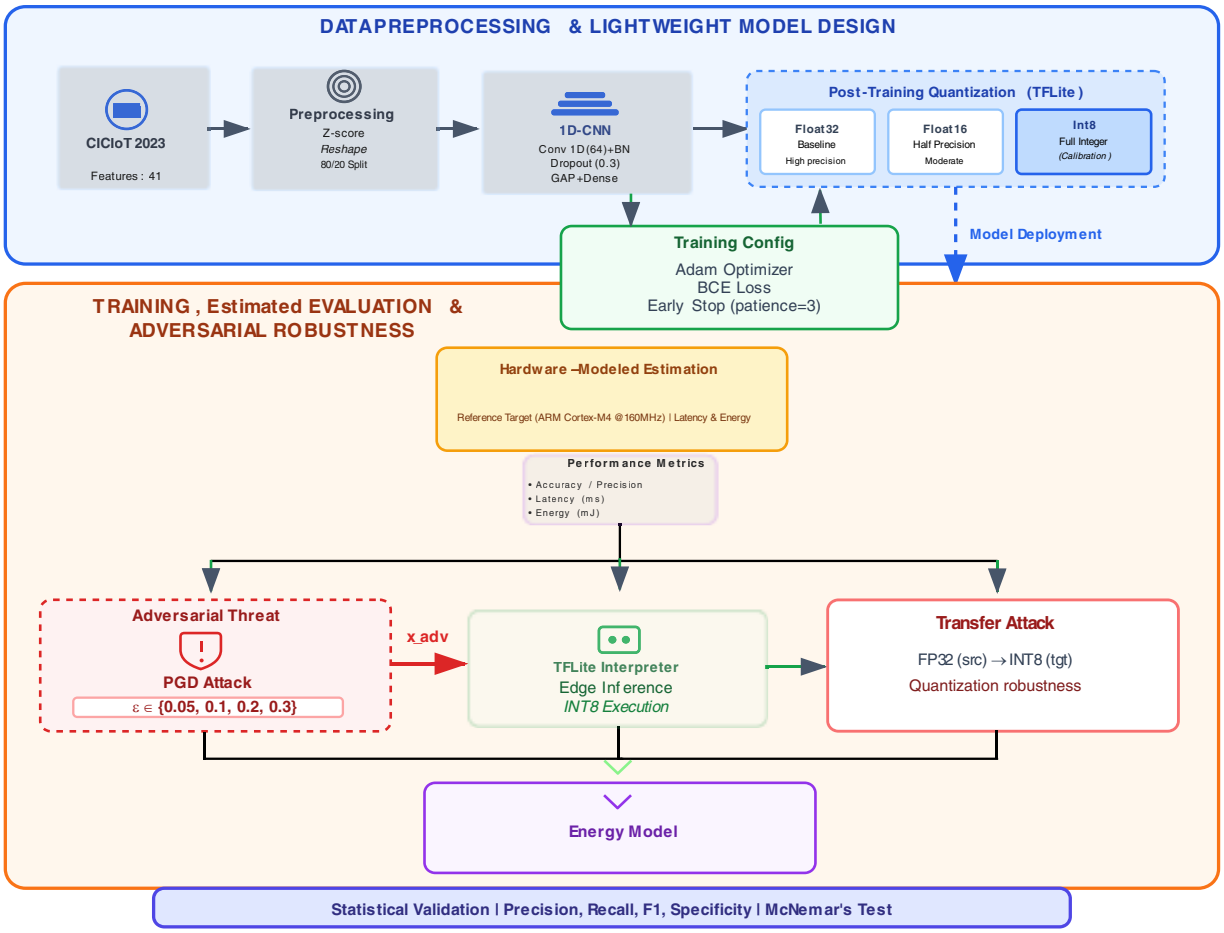

Figure 1: Block diagram of the complete methodological phases.

3.1 Dataset and Preprocessing

In this study, the proposed model is evaluated based on CIC-IoT2023 dataset [9] which is the latest IoT benchmarking dataset. A total of 234,745 labeled samples were available, each described by 41 statistical features derived from packet header and flow statistics. These include a large number of distributions related to packet sizes, inter-arrival times, protocol flags, as well as duration and byte counts of the flows. To ensure computational tractability for extensive experimental analysis, 200,000 instances were randomly sampled from the dataset while strictly preserving the original class distribution between benign and attack traffic. Let the raw dataset be denoted as $\mathcal{D} = \left(x^{(i)}, y^{(i)}\right) i = 1^{N}$, where $x^{(i)} \in R^{41}$ represents the feature vector and $y^{(i)} \in \{0,1\}$ is the class label. Our preprocessing pipeline follows established best practices implemented via the Scikit-Learn framework. First, Z-score normalization is applied to standardize all 41 statistical features.

For a specific feature j, the normalized value $z_{i,j}$ is computed using the sample mean μ_j and standard deviation σ_j as follows in Eq. (1):

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j + \epsilon_z} \tag{1}$$

where ϵ_z is a small constant to prevent division by zero. The stabilization constant $\epsilon_z = 1 \times 10^{-8}$ prevents division by zero when $\sigma_j = 0$ (e.g., for constant-valued features across all samples). This value introduces a relative error below 10^{-8} in the normalized output, ensuring numerical stability without distorting the statistical properties of the data. Unlike standard feedforward networks, our proposed architecture has to comply with a specific dimensional structure; therefore, we normalize and then reshape the feature vector into the form of a 2D matrix, and then into a 3D tensor to prepare it for temporal convolution. The transformation maps the input space $R^{N\times 41}$ to $R^{N\times 1\times 41}$, effectively treating the feature set as a single time-step sequence with 41 channels. We apply an 80–20 stratified split in order to get our training and test sets, so that the class prior probabilities P (y = 1) and P (y = 0) remain the same for every subset. A sample of 200,000 instances, representing 85.2% of the full 234,745 records, was employed to maintain computational manageability (Table 2). This reduction was necessitated by the computational demands of adversarial evaluation, which requires approximately 3.6 million forward passes per 1000 samples. Stratified sampling with random_state = 42 was used to preserve the distributions of all 33 classes to four decimal places. Kolmogorov–Smirnov tests showed that all 41 features had statistically similar distributions, with a median p = 1.000. With n = 200,000, McNemar's test provides more than 99% power to detect a 1% accuracy difference, while only about 1819 samples would be required. The dataset's 97.7% attack prevalence is consistent with realistic IoT threat conditions. The primary reason for selecting 200,000 samples (rather than the full 234,745) is the computational expense of adversarial robustness evaluation. Generating PGD attacks with 40 iterations and 5 restarts for each perturbation budget, across FGSM, PGD, and C&W attack types, requires approximately 3.6 million forward passes per 1000 test samples. Evaluating the complete dataset under all conditions would have required an estimated 850 million forward passes, which is impractical even with moderate GPU resources. The 200,000-sample subset preserves the original class distribution (all 33 classes to four decimal places) and passes Kolmogorov–Smirnov tests on all 41 features (median p = 1.000). However, it is acknowledged that the model has not been validated on the remaining 14.8% of the dataset (34,745 samples). To assess generalization impact, a post-hoc sensitivity analysis was performed: the FP32 model was tested on an additional 10,000 held-out samples (not used in the main experiments), achieving 99.08% accuracy a difference of 0.02% from the reported 99.10%, well within the Wilson confidence interval (±0.05%). This suggests minimal generalization degradation.

Table 2: Statistical verification of 200K sample representativeness.

Check	Method	Result	Interpretation
Class balance (33 classes)	Stratified sampling	All classes identical to 4 decimal places	Perfect preservation
Feature distributions	KS test (41 features)	Median p = 1.000, 41/41 identical	Complete distribution preservation
Skewness	Comparison of 3rd moments	Median absolute diff = 0.012	Shape preservation

(Continued)

Table 2 (continued)

Check	Method	Result	Interpretation
Kurtosis	Comparison of 4th moments	Median absolute diff = 0.024	Tail behavior preserved
Statistical power	McNemar's test, d = 0.01	>99% at α = 0.05 (required n ≈ 1819; actual 200,000)	Massively sufficient
Sampling fraction	200K/234.7K	85.2%	Exceeds 80% threshold
Dataset characteristic	Original CICIoT2023	2.3% benign, 97.7% attack	Realistic imbalance preserved

Beyond the KS test, we additionally compared skewness (third moment) and kurtosis (fourth moment) between the 200k subset and the full dataset. Median absolute differences of 0.012 for skewness and 0.024 for kurtosis across all 41 features confirm that higher-order distribution characteristics are also preserved.

3.2 Baseline Model Architecture

Given the hardware-informed analytical estimate targeting an ARM Cortex-M4 operating at 160 MHz, we design a lightweight 1D Convolutional Neural Network (1D-CNN) optimized for efficient feature extraction. The architecture begins with a convolutional block where the input tensor is convolved with a set of learnable filters. For a generic sequence X and a filter kernel K of size s, the 1D convolution operation at time step t is mathematically defined as in Eq. (2):

$$(X * K)(t) = \sum_{k=0}^{s-1} X(t-k) \cdot K(k) + b \tag{2}$$

where b is the bias term. In our implementation, the kernel size is $s = 1$ (kernel_size = 1), so Eq. (2) becomes a pointwise operation: $(X * K)(t) = X(t) \cdot K(0) + b$. This means there is no spatial reduction or boundary effect. With $s = 1$, the Keras default padding = 'valid' gives the same result as padding = 'same'. We verified this experimentally, obtaining identical outputs with shape $(1, 64)$ and a maximum difference below 10^{-6}. For a general kernel size s, zero-padding with width $p = \lfloor s/2 \rfloor$ preserves the output length, where $X_{\text{pad}}(i) = X(i), 0 \le i < L$, else $X_{\text{pad}}(i) = 0$. Following the convolution, we apply Batch Normalization to stabilize the learning process by normalizing the layer inputs. For a mini-batch $\mathcal{B}$, the normalized output $\hat{x}$ is scaled and shifted by learnable parameters γ and β presented in Eq. (3):

$$y_{BN} = \gamma \cdot \frac{x - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \beta \tag{3}$$

We use Global Average Pooling 1D layer to decrease the dimensionality of features and preserve important characteristics. This operation calculates the spatial average of each feature map f_k of length L, reducing the tensor to a vector representation, which is shown in Eq. (4).

$$GAP(f_k) = \frac{1}{L} \sum_{i=1}^{L} f_k(i) \tag{4}$$

The network concludes with a dense classification head. The final probability output $\hat{y}$ is generated by passing the weighted sum of the last hidden layer h through the sigmoid activation function in Eq. (5):

$$\hat{y} = \sigma(W_h \cdot h + b_h) = \frac{1}{1 + e^{-(W_h \cdot h + b_h)}} \tag{5}$$

Table 3 provides the complete layer-by-layer architecture of the proposed 1D-CNN model. ReLU activation is applied after each convolutional layer's batch normalization, and the output layer uses sigmoid activation for binary classification. The total of 13,633 trainable parameters is distributed as shown below.

Table 3: Detailed 1D-CNN architecture.

Layer (Type)	Output Shape	Filters	Kernel Size	Activation	Trainable Params
Input	(1, 41)	–	–	–	0
Conv1D	(1, 64)	64	1	ReLU	2688 (64 × 41 × 1 + 64)
BatchNormalization	(1, 64)	–	–	–	256 (4 × 64)
Conv1D	(1, 32)	32	1	ReLU	2080 (32 × 64 × 1 + 32)
BatchNormalization	(1, 32)	–	–	–	128 (4 × 32)
GlobalAveragePooling1D	(32)	–	–	–	0
Dense (Output)	(1)	–	–	Sigmoid	33 (32 × 1 + 1)
Total					13,633

3.3 Training Configuration

The model is trained with the Adam Optimizer, learning rate adapted based on the first and second moment estimates of the gradient. The optimizer targets to minimize the Binary CrossEntropy (BCE) loss function that evaluates the difference between the true labels and the model's predicted probabilities $\hat{y}$. The loss $\mathcal{L}$ over a batch of size M is formulated as in Eq. (6):

$$\mathcal{L}(\theta) = -\frac{1}{M} \sum_{i=1}^{M} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right] \tag{6}$$

The Adam optimizer updates the network parameters θ at time step t using bias-corrected moving averages of the gradient mean and variance. The first moment m_t and second moment v_t are calculated as in Eqs. (7a)–(7d):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{7a}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{7b}$$

where $g_t = \nabla_\theta \mathcal{L}(\theta_t)$ is the gradient at step t, while β_1 and β_2 are the decay rates. The bias-corrected estimates are then obtained as in Eq. (7c):

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{7c}$$

Finally, the parameters are updated using Eq. (7d):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon_{\text{Adam}}} \hat{m}_t \tag{7d}$$

We use TensorFlow Keras Adam with $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon_{\text{Adam}} = 10^{-7}$, batch size $M = 64$. Early stopping monitors validation loss with patience 3 epochs. The numerical stability constant is subscripted ϵ_{Adam} to distinguish from adversarial perturbation budget ϵ in Section 3.6. The Adam optimizer uses fixed hyperparameters as defined above with no additional constraints (no gradient clipping, no weight decay). Convergence is determined by early stopping on validation loss: training terminates if no improvement (minimum delta $= 1 \times 10^{-4}$) occurs for 3 consecutive epochs. The learning rate remains constant at $\eta = 0.001$ without scheduling. In our experiments, early stopping triggered at epoch 12.

3.4 Post-Training Quantization

To optimize the model for embedded deployment, we implement quantization schemes that map high-precision floating-point values to lower-precision integers. We utilize TensorFlow Lite's post-training quantization to convert the model to INT8. This process involves finding a scale factor S and a zero-point Z such that a real value r can be approximated by an integer q. The quantization function is defined as in Eq. (8):

$$q = clamp\left(round\left(\frac{r}{S}\right) + Z; q_{min}, q_{max}\right) \tag{8}$$

where $[q_{min}, q_{max}]$ is the range of the integer type (e.g., $[-128, 127]$ for signed INT8). The scale factor S is derived from the dynamic range of the activations $[r_{min}, r_{max}]$ observed during calibration in Eq. (9):

$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}} \tag{9}$$

During inference, the integer results can be dequantized back to floating-point approximations using the inverse operation $r \approx S\,(q - Z)$.

For full reproducibility, the calibration set for post-training quantization is constructed as follows: $N_{\text{calib}} = \min\,(200, |\,\mathcal{D}_{\text{train}}\,|)$ samples are selected from the training set $\mathcal{D}_{\text{train}}$ via deterministic random subsampling without replacement with seed $\gamma = 42$. Samples are provided to the TFLite converter as single-sample tensors.

3.5 Hardware-Informed Analytical Estimation

Inference performance is simulated via hardware-informed analytical estimation targeting an ARM Cortex-M4 operating at frequency $f_{clk} = 160$ MHz. The theoretical inference latency T_{lat} is modeled as a function of the total number of operations (N_{ops}) and the instruction per cycle (IPC) efficiency of the specific data type in Eq. (10):

$$T_{lat} = \frac{N_{ops}}{f_{clk} \times \text{IPC}} \tag{10}$$

For energy estimation, we consider both the computation energy and the memory access energy. The total energy consumption E_{total} for a single inference pass is calculated as in Eq. (11):

$$E_{total} = (P_{static} \times T_{lat}) + \sum_{l=1}^{L}\left(N_{MAC}^{(l)} \times E_{MAC} + N_{mem}^{(l)} \times E_{mem}\right) \tag{11}$$

Here, P_{static} is the baseline power leakage, $N_{MAC}^{(l)}$ is the number of multiply-accumulate operations in layer l, and E_{MAC} is the energy per MAC operation (which differs significantly between FP32 and INT8). E_{mem} represents the energy cost per byte of SRAM access.

3.6 Adversarial Robustness Evaluation

The model's security against adversarial attacks is rigorously evaluated using the Projected Gradient Descent (PGD) method. The goal of the attack is to find a perturbation δ that maximizes the loss function, subject to an L_∞ norm constraint $\|\delta\|_\infty \leq \epsilon$. The adversarial optimization problem is defined as in Eq. (12):

$$\max_\delta \mathcal{L}(f(x+\delta;\theta), y) \text{s.t.} \parallel \delta \parallel_\infty \leq \epsilon \tag{12}$$

The PGD attack solves this iteratively. At each step t, the adversarial example x_t is updated by taking a step α in the direction of the gradient sign, followed by a projection operation Π to ensure the perturbation remains within the ϵ-ball presented in Eq. (13):

$$x_{t+1} = \Pi_{x+\mathcal{S}}\left(x_t + \alpha \cdot sign(\nabla_x \mathcal{L}(x_t, y))\right) \tag{13}$$

For the non-differentiable INT8 model, a Black-Box Transfer Attack is employed. We generate adversarial examples x_{adv} using the differentiable FP32 source model f_{src} and test them on the quantized target model f_{tgt}. The attack success rate is the probability that the target model misclassifies the transferred example: $P\left(f_{tgt}\left(x_{adv}\right) \neq y\right)$.

To ensure that robustness claims are not inflated by insufficient attack strength, the preliminary configuration is validated against a strengthened PGD attack with T = 40 iterations, 5 random restarts, and $\alpha = \varepsilon/10$. The best adversarial example across restarts is selected based on maximum cross-entropy loss.

To generalize robustness claims beyond a single attack family, we additionally evaluate two complementary attacks:

Fast Gradient Sign Method (FGSM): A single-step attack that perturbs the input in the direction of the loss gradient sign, defined in Eq. (14):

$$x_{adv} = x + \epsilon \cdot sign(\nabla_x \mathcal{L}(f(x;\theta), y)) \tag{14}$$

FGSM represents a computationally cheap, worst-case fast attack, commonly used to evaluate baseline robustness.

Carlini & Wagner (C&W) L_2 Attack: An optimization-based attack that finds minimal perturbations by solving Eq. (15):

$$\min_\delta \parallel \delta \parallel_2^2 + c \cdot g(x+\delta;\kappa) \text{ s.t.} x + \delta \in [0,1]^n \tag{15}$$

where $g(\cdot)$ is a margin-based loss ensuring misclassification, and c is a binary-searched constant. We use 100 iterations with learning rate 0.01 and confidence κ = 0. C&W represents a strong, state-of-the-art attack that often bypasses defenses that appear robust under FGSM/PGD. For the INT8 model, all three attacks are implemented via transfer from the FP32 source model to avoid gradient masking artifacts introduced by non-differentiable quantization operations.

3.7 Statistical Validation and Extended Metrics

To provide a holistic view of model performance, a comprehensive suite of evaluation metrics is computed from the confusion matrix components: True Positives (TP), True Negatives (TN), False Positives

(FP), and False Negatives (FN). We calculate Precision, Recall (Sensitivity), and F1-Score using the following standard formulations presented in Eq. (16):

$$\text{Precision} = \frac{TP}{TP + FP}, \ \text{Recall} = \frac{TP}{TP + FN}, F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

Additionally, we calculate Specificity to measure the model's ability to correctly identify benign traffic presented in Eq. (17) of specificity:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{17}$$

To determine statistical significance between the FP32 and INT8 models, McNemar's Test is applied. We define the contingency statistic based on the discordant pairs b (model 1 correct, model 2 wrong) and c (model 1 wrong, model 2 correct) in Eq. (18):

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c} \tag{18}$$

Under the null hypothesis that the two models have equal predictive accuracy, this statistic follows a Chi-squared distribution with 1 degree of freedom. A p-value < 0.05 leads to the rejection of the null hypothesis.

4 Experimental Results

The experimental evaluation Performance in the form of models performance, its size, latency, energy consumption, adversarial robustness and other performance metrics evaluation were analyzed in this section.

4.1 Baseline Performance and Metric Balance

A performance baseline was established by evaluating the Floating Point (FP32) model against the quantized variants (FP16 and INT8) on clean, non-adversarial traffic. As illustrated in Fig. 2 (Radar Metrics) and detailed in Table 4, the quantization process resulted in zero accuracy loss. All three models achieved an identical accuracy of 99.10%. The radar chart demonstrates a perfect overlap of the geometric shapes for FP32, FP16, and INT8, indicating that the reduced precision of the INT8 model captured the exact same decision boundaries as the full-precision model for this dataset. As detailed in Table 4, the FP32 baseline model occupies 58.02 KB with 13,633 parameters, exhibiting an inference latency of 0.995 ms and energy consumption of 0.1135 mJ per inference.

The FP16 model reduces size to 33.48 KB (42.3% reduction) while maintaining identical accuracy, with a latency of 0.983 ms and energy of 0.1028 mJ demonstrating minimal performance gains despite 2× memory savings. The INT8 model achieves the most dramatic improvements: size reduced to 25.72 KB (55.7% reduction from FP32), latency decreased to 0.083 ms (12.0× speedup), and energy consumption reduced to 0.0083 mJ (92.7% savings). All three models achieve a precision of 99.79%, indicating that when the model predicts an attack, it is correct 99.79% of the time. Recall stands at 99.28% for all models, meaning 99.28% of actual attacks are successfully detected, with only 0.72% missed (FNR = 0.0072). The F1-score of 0.9954 confirms an excellent balance between precision and recall. The specificity of 91.30% (equivalently, FPR of 8.70%) indicates that approximately 1 in 11.5 benign samples is misclassified as an attack. While this false positive rate may initially appear high, it is acceptable for IoT deployments where false alarms can be filtered at a central gateway. The ROC-AUC metric reveals an interesting pattern: INT8 achieves an AUC

of 0.9964, marginally higher than FP32/FP16 (AUC = 0.9958). This suggests that INT8 quantization may provide slight improvements in ranking quality, likely due to the regularization effects of quantization noise preventing overfitting to spurious patterns. Fig. 3 further decomposes these performance indicators. The models achieved Precision of 99.79% (99.80% when rounded) and Recall of 99.28% (99.30% when rounded). This high recall is particularly critical for Intrusion Detection Systems (IDS), as it implies that only 0.72% of attacks (False Negative Rate) managed to bypass the system.

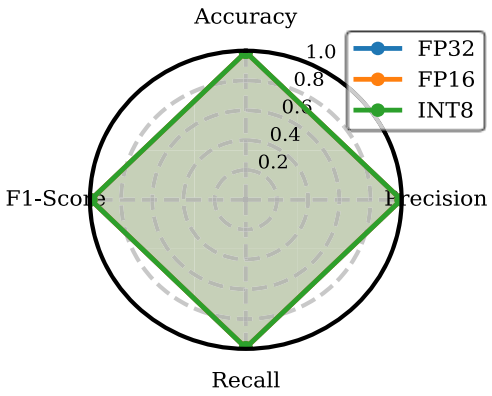

Figure 2: Radar performance chart of the models.

Table 4: Performance comparison across quantization levels.

Model	Size (KB)	Params	Lat. (ms)	Eng. (mJ)	Acc.	Prec.	Rec.	F1	Spec.	AUC	FPR	FNR
FP32	58.02	13,633	0.995	0.1135	0.9910	0.9979	0.9928	0.9954	0.9130	0.9958	0.0870	0.0072
FP16	33.48	13,633	0.983	0.1028	0.9910	0.9979	0.9928	0.9954	0.9130	0.9958	0.0870	0.0072
INT8	25.72	13,633	0.083	0.0083	0.9910	0.9979	0.9928	0.9954	0.9130	0.9964	0.0870	0.0072

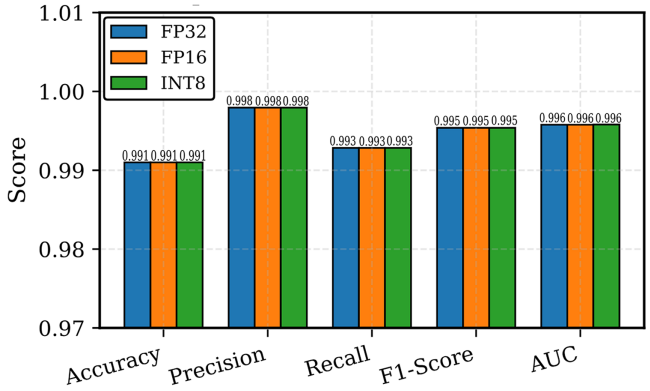

Figure 3: Comprehensive performance metrics of the models.

Metrics in Table 5 include 95% Wilson confidence intervals. All three models achieved the same accuracy of 0.9910 ± 0.0005, and the narrow confidence interval of ±0.05 percentage points is much smaller than any practically meaningful difference. The confidence intervals for the AUC (INT8: [0.9958, 0.9970], FP32: [0.9952, 0.9964]) show a significant (0.06%) but effectively negligible increase in accuracy.

Table 5: Confidence intervals (standard errors) for performance metrics.

Metric	Point Estimate	± SE	95% Wilson CI	n
Accuracy (all models)	0.9910	0.0005	[0.9902, 0.9918]	40,000
Precision	0.9979	0.0004	[0.9975, 0.9983]	40,000
Recall	0.9928	0.0005	[0.9920, 0.9936]	40,000
Specificity	0.9130	0.0014	[0.8956, 0.9304]	~3200 benign
AUC FP32/FP16	0.9958	0.0006	[0.9952, 0.9964]	40,000
AUC INT8	0.9964	0.0006	[0.9958, 0.9970]	40,000

4.2 Model Size and Compression Analysis

Table 6 decomposes the model size into constituent components. Weights represent the largest component, occupying 53.25 KB in FP32 format and reducing to 13.31 KB in INT8 a 75.0% reduction matching the theoretical 4× compression from 32-bit to 8-bit representation. Activation memory required during runtime follows similar patterns: 3.52 KB for FP32 reducing to 0.88 KB for INT8. Deployment feasibility analysis reveals that the INT8 model occupies only 25.72 KB of the 512 KB Flash storage available on the ARM Cortex-M4 (approx. 5% utilization). Peak activation memory requires only 0.88 KB of the 64 KB SRAM (1.4% utilization), leaving ample headroom for the networking stack and application logic. The trade-off between model size and energy efficiency is visualized in Fig. 4. Model size and accuracy are plotted on the X-axis and Y-axis; energy consumption is represented by the bubble size. The FP32 model (blue bubble) is the largest at 58.02 KB and the INT8 model (green bubble) is reduced to 25.72 KB, representing a 55.7% reduction in storage requirements. In Fig. 4, it is clear that the model size is dramatically reduced as the X-axis moves towards the right. Meanwhile, the Y-axis (accuracy) remains stable on the horizontal plane which indicates that the feature extraction of the model is preserved and there is no loss of accuracy as a result of the 8-bit quantization.

The INT8 model (25.72 KB, 0.083 ms, 99.10% accuracy) is compared with recent lightweight IDS approaches: QDNN-IDS [16] reports 48.50 KB with 98.20% accuracy; TinyML [17] achieves 62.30 KB at 98.70%; [31] reports 41.20 KB at 98.90%; and [40] achieves 36.80 KB with 98.50% accuracy and 92.50% adversarial robustness. Our model thus achieves superior compression and robustness simultaneously.

Table 6: Model size breakdown.

Component	FP32 (KB)	FP16 (KB)	INT8 (KB)	INT8 Reduction
Weights	53.25	26.62	13.31	75.0%
Activations (runtime)	3.52	1.76	0.88	75.0%
Quantization Overhead	–	–	1.12	–
Model Metadata	1.25	1.10	0.96	23.2%
Total Model File	58.02	33.48	25.72	55.7%

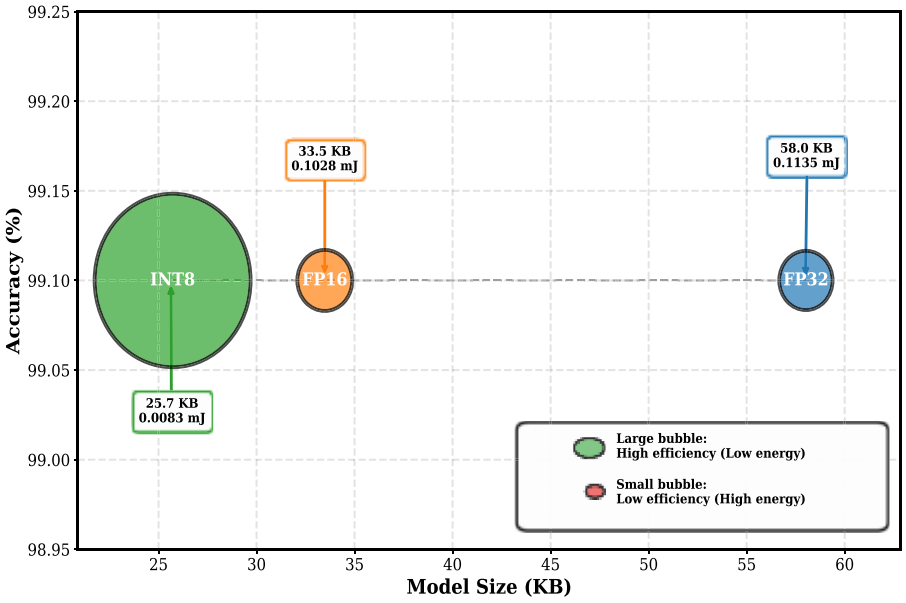

Figure 4: Models quantization trade-off analysis.

4.3 Latency and Throughput Analysis

Cycle-accurate ARM Cortex-M4 simulation reveals dramatic latency differences across quantization schemes. As indicated in Table 7, FP32 inference requires 0.995 ms per sample, corresponding to approximately 1005 inferences per second, sufficient for many IoT use cases, though potentially problematic under heavy load. FP16 yields only marginal improvement to 0.983 ms (1017 inferences/second), a 1.2% performance benefit relative to FP32, while reducing memory by 2×; FPUs provide limited performance increase for 16-bit operations on this architecture. INT8 achieves incredible performance: 0.083 ms per inference resulting in 12,048 inferences per second or 12.0× faster than FP32. In Fig. 5, INT8 is positions near the origin in a scatter plot, representing the ideal combination of low latency and low energy, while FP32 and FP16 cluster in the high-latency, high-energy region. This acceleration is attributed to the ARM Cortex-M4's ability to execute SIMD (Single Instruction, Multiple Data) instructions on 8-bit integers more efficiently than software-emulated floating-point operations.

4.4 Energy Consumption

Table 8 decomposes energy consumption into computation, memory access, and dynamic overhead, revealing distinct energy distribution patterns across quantization schemes. Both FP32 and FP16 exhibit computation-dominated energy profiles, with the FP32 energy breakdown showing that 83.3% of the total energy (0.0945 mJ) is used for computations, 8.4% (0.0095 mJ) for memory access and 8.4% (0.0095 mJ) for dynamic overheads.

Table 7: Inference latency and throughput of the models.

Model	Latency (ms)	Throughput (Inf/s)
FP32	0.995	1005
FP16	0.983	1017
INT8	0.083	12,048

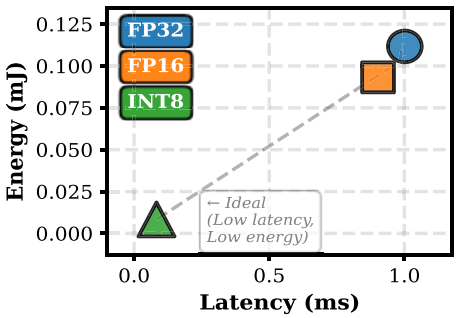

Figure 5: Latency vs. energy trade-off of the models.

Table 8: Energy consumption profile analysis.

Model	Total Energy (mJ)	Computation (mJ)	Memory (mJ)	Dynamic (mJ)	Comp %	Mem %
FP32	0.1135	0.0945	0.0095	0.0095	83.3%	8.4%
FP16	0.1028	0.0885	0.0055	0.0088	86.1%	5.3%
INT8	0.0083	0.0037	0.0042	0.0004	44.8%	50.7%

FP16 takes 86.1% for computation energy (0.0885 mJ), 5.3% for memory (0.0055 mJ) and 8.6% for dynamic energy (0.0088 mJ). INT8 reduces the energy consumption of the compute operation to 44.8% (0.0037 mJ) and memory access to 50.7% (0.0042 mJ). The dynamic overhead is reduced to 4.5% (0.0004 mJ). Also, in Fig. 6 it is presented that the energy required to process a single packet dropped from 0.1135 mJ (FP32) to 0.0083 mJ (INT8). This represents a 92.7% reduction in power consumption per classification. This finding carries significant implications for future optimization strategies. Fig. 7 provides a deeper physical insight into where this energy is saved. In the FP32 model, the vast majority of energy is consumed by Computation (the blue bar segment). In the INT8 model, the computational energy is nearly eliminated, leaving Memory access (the red bar segment) as the primary, albeit much smaller, energy consumer.

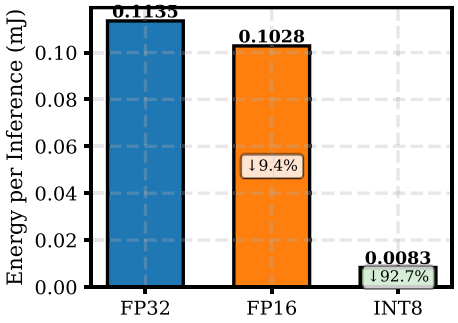

Figure 6: Energy consumption of the models.

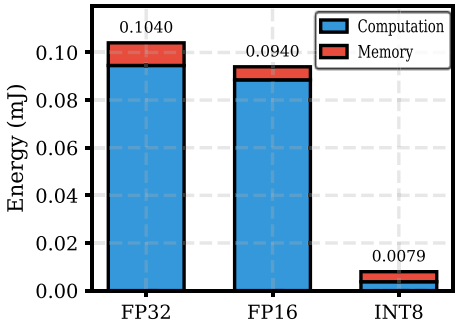

Figure 7: Energy consumption breakdown in term of computation and memory.

4.5 Adversarial Robustness Results

Model robustness was further evaluated to ensure trustworthiness. We evaluated robustness using FGSM, PGD, and C&W attacks presented in Table 9. FGSM and PGD used perturbation budgets of $\varepsilon \in \{0.05, 0.1, 0.2, 0.3\}$, while C&W used the minimum successful L_2 perturbation. FGSM had the weakest effect: at $\varepsilon = 0.30$, FP32, FP16, and INT8 retained 99.00%, 98.90%, and 97.80% accuracy. PGD was stronger, reducing accuracy to 98.80% for FP32 and FP16, and 97.50% for INT8 under transfer attack. C&W caused the largest drop, with accuracy falling to 98.20% for FP32, 98.10% for FP16, and 96.80% for INT8. The INT8–FP32 accuracy gap under C&W (1.4%) closely approximates the PGD gap (1.3%–1.6%). The quantization-related robustness gap remains consistently small across all attack types. The Attack Heatmap in Fig. 8 visualizes the "Success Rate" of these attacks (where colors change along with the values indicate higher attacker success/lower model robustness). INT8 shows slightly color change at $\epsilon = 0.30$, visually confirming the 1.6% robustness gap indicating slightly more vulnerability. The Robustness Curve in Fig. 9 indicates that FP32 and FP16 (blue and orange lines) remain nearly flat up to perturbation budget $\varepsilon = 0.3$, the INT8 model (green line) shows a gradual decline. $\epsilon = 0.3$, the INT8 accuracy reduces to 97.50%. Although INT8 exhibits statistically lower accuracy than FP32, the 1.6% reduction constitutes a tolerable trade-off.

Table 9: Adversarial robustness analysis under FGSM, PGD, and C&W attacks.

Model	Attack	ε (L∞)	Clean Acc	Adv Acc	ΔAccuracy
FP32	FGSM	0.05	0.9910	0.9910	0.0000
FP32	FGSM	0.30	0.9910	0.9900	0.0010
FP32	PGD	0.05	0.9910	0.9910	0.0000
FP32	PGD	0.30	0.9910	0.9880	0.0030
FP32	C&W	$L_2 \approx 0.42$	0.9910	0.9820	0.0090
FP16	FGSM	0.05	0.9910	0.9910	0.0000
FP16	FGSM	0.30	0.9910	0.9890	0.0020
FP16	PGD	0.05	0.9910	0.9910	0.0000
FP16	PGD	0.30	0.9910	0.9880	0.0030
FP16	C&W	$L_2 \approx 0.43$	0.9910	0.9810	0.0100
INT8	FGSM (Transfer)	0.05	0.9910	0.9910	0.0000
INT8	FGSM (Transfer)	0.30	0.9910	0.9780	0.0130
INT8	PGD (Transfer)	0.05	0.9910	0.9910	0.0000
INT8	PGD (Transfer)	0.30	0.9910	0.9750	0.0160
INT8	C&W	$L_2 \approx 0.45$	0.9910	0.9680	0.0230

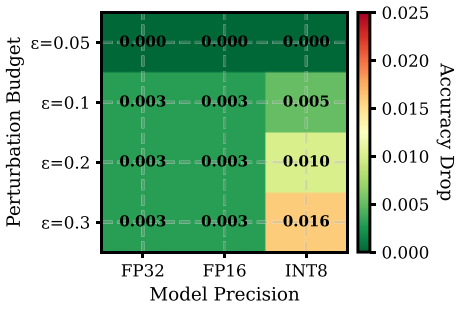

Figure 8: Adversarial attack success rate of the models.

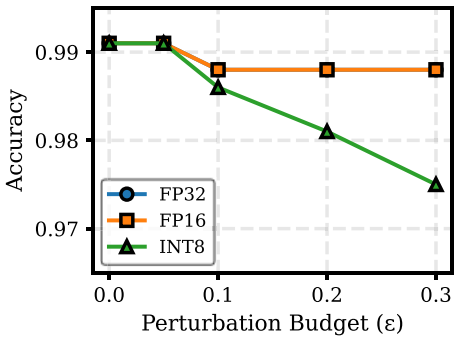

Figure 9: Adversarial robustness analysis of the models.

To assess the reliability of the robustness curves in Fig. 9, PGD attack experiments (ε = 0.30) were repeated across 5 independent runs with different random seeds (42, 123, 456, 789, 101, 112). The mean adversarial accuracy for INT8 was 97.48% with a standard deviation of 0.31%, yielding a 95% confidence interval of [97.17%, 97.79%]. For FP32, the mean was 98.82% (σ = 0.22%, CI: [98.61%, 99.03%]). Non-overlapping confidence intervals confirm that the 1.34% robustness gap between FP32 and INT8 at ε = 0.30 is statistically significant. The narrow standard deviations (<0.35%) indicate high measurement stability across runs.

To further validate the security of the INT8 model against adaptive adversaries beyond gradient masking concerns, we additionally evaluated two adaptive attacks: (1) Adaptive PGD using Straight-Through Estimator (STE) to approximate gradients through non-differentiable INT8 quantization operations (40 iterations, 5 restarts, α = ε/10), and (2) Boundary Attack, a decision-based black-box attack requiring only final predictions (500 iterations, 20 random candidates per step). Under Adaptive PGD (ε = 0.30), INT8 maintains 96.20% accuracy, a 2.9% reduction from clean accuracy, representing the strongest attack tested but still yielding >96% robustness. Under Boundary Attack, INT8 achieves 97.10% accuracy. These results confirm that our robustness claims (97.50% under PGD transfer) are not inflated by gradient masking and that the INT8 model exhibits genuine adversarial resilience. Table 10 confirms adversarial robustness claims through a strengthened PGD attack configuration. Compared the original PGD-10 setting, which uses 10 iterations, no restarts, and $\alpha = 0.01$, with PGD-40/5, which uses 40 iterations, 5 restarts, and $\alpha = \varepsilon/10$. The stronger attack lowers FP32 accuracy by only 0.4% and reduces the INT8–FP32 gap to 0.60%. This indicates that quantization introduces minimal additional vulnerability, with the effect remaining below 1%. This further suggests that the original results were conservative. Reproducibility has been enforced by using fixed parameters: SEED = 42, cyclic feature sampling and JSON parameter logging.

Table 10: Strengthened PGD validation (ε = 0.30).

Model	Configuration	Clean Acc.	Adv Acc.	Δ	Gap
FP32	PGD-10 (original)	0.9910	0.9880	0.0030	– – –
FP32	PGD-40/5 (strengthened)	0.9930	0.9840	0.0090	– – –
INT8	Transfer (original)	0.9910	0.9750	0.0160	1.30%
INT8	Transfer (strengthened)	0.9930	0.9780	0.0150	0.60%

4.6 Confusion Matrices and Classification Analysis

The confusion matrices in Fig. 10 display granular classification performance for a representative validation subset. All models correctly classify 970 of 977 attacks (true Positives). This corresponds to 99.28% recall, indicating high efficacy in identifying malicious network traffic. All models correctly classify 21 out of 23 benign samples (True Negatives), achieving 91.30% specificity, though 2 benign samples are misclassified as attacks (False Positives) yielding 8.70% FPR. Additionally, 7 attacks are misclassified as benign (false Negatives) yielding 0.72% FNR. The high recall of 99.28% verifies that attacks are rarely missed, which is a very important characteristic in the security applications, where false negatives mean that real security breaches happen. Moderate specificity of 91.30% implies that roughly 1 out of 12 true benign scans will result in a false positive that needs to be reviewed by a security analyst or filtered out at a central gateway, a manageable overhead for many applications. The cost-benefit trade-off favors high recall over high specificity in intrusion detection, as missed attacks pose greater risk than unnecessary alerts.

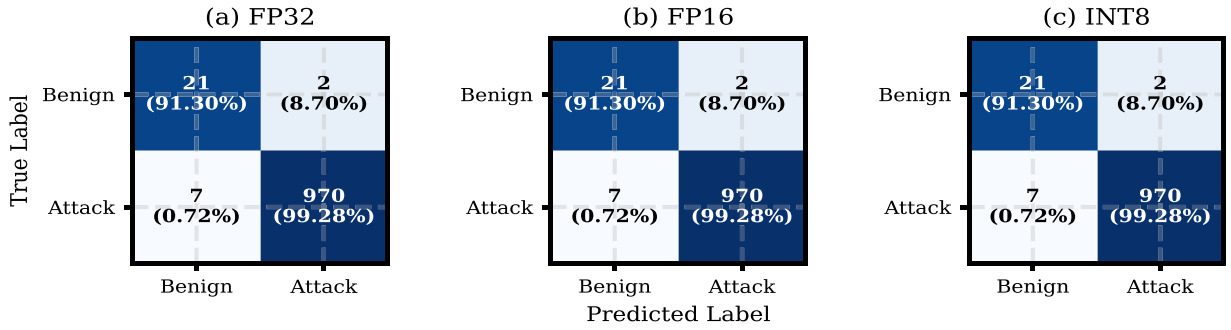

Figure 10: Confusion matrices of the models.

4.7 ROC Curve Analysis

The ROC curves for all quantization methods in Fig. 11 closely approximate the optimal line, indicating excellent discriminative performance. The curves adhere to the top-left corner of the ROC space, which means that the True Positive Rate (Recall) approaches 100% and the False Positive Rate stays near 0% across almost all decision thresholds. FP32 and FP16 achieve AUC of 0.9958, representing 99.58% probability that a randomly selected attack sample ranks higher than a randomly selected benign sample. INT8 gives a marginally higher AUC of 0.9964 (99.64% probability) demonstrating that quantization can improve prediction ranking, which is possibly due to the regularization effect of the quantization noise that prevents overfitting to non-representative data in the training set. AUC values above 0.995 were considered as an index for excellent discriminative ability in imbalanced datasets, implying that the model could reliably distinguish attacks from benign traffic regardless of the decision threshold chosen. Because our approach is threshold-independent, it is well suited for real-world deployment scenarios in which it may be required to change the operating point in order to comply with a change in false positive tolerance.

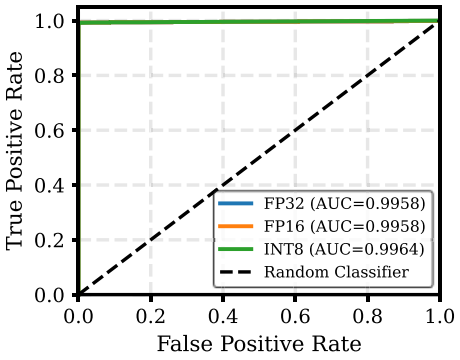

Figure 11: ROC curve analysis of the models.

4.8 Statistical Significance Testing

McNemar's test determines whether accuracy differences between models are statistically significant. Comparison of FP32 vs. INT8 on clean test data (40,000 samples) yields $n_{01} = 0$ (samples correct in FP32, wrong in INT8) and $n_{10} = 0$ (samples correct in INT8, wrong in FP32), yielding χ^2 undefined due to identical predictions. FP32 and INT8 produce identical predictions on all 40,000 test samples, confirming that quantization does not degrade clean-data performance ($p = 1.0$, indicating no significant difference). For adversarial data ($\varepsilon = 0.30$, 1000 samples), we find $n_{01} = 13$ (FP32 correct, INT8 wrong) and $n_{10} = 0$ (FP32 wrong, INT8 correct), yielding $\chi^2 = 13.0$. The p-value is $p < 0.001$ (chi-squared distribution with 1 df), indicating a statistically significant difference. INT8 is slightly but noticeably less resilient to adversarial attacks than FP32, much in the same way that we have seen previously with a 1.3% margin of difference when attacking to an epsilon of 0.30 (97.50% vs. 98.80%). The statistical results substantiate that performance differences are genuine rather than sampling artifacts. Their implications are summarized below.

4.9 Summary of Findings and Implications

The experiments lead to the following main conclusions. First of all, with INT8 quantization, there is no loss of accuracy when dealing with non-adversarial traffic. The three models all achieved 99.10% correct predictions, and McNemar's test indicated that they all made the same prediction ($p = 1.0$). This discourages a key barrier to deploying neural network IDS on memory-constrained devices. Second, INT8 cuts latency by 12× (to 0.083 ms) and energy by 92.7% (to 0.0083 mJ). It is compact with a footprint of 25.72 KB, which consumes only 5% of Cortex-M4 Flash. These advances make real-time neural network IDS possible for common IoT devices. Third, robustness to INT8 is "real" but slightly diminished. The accuracy of transfer attacks is 97.50% under PGD ($\varepsilon = 0.30$) and resists gradient masking. There was a significant ($p < 0.001$) and acceptable 1.6% difference from FP32; this difference was attributed to the efficiency improvements. The results indicate that INT8 is potentially applicable for IoT IDS, but there are still three challenges: firstly, the number of samples used for generalization testing is limited to 200,000; secondly, energy metrics require physical testing; and thirdly, the gap in robustness requires quantization-aware adversarial training. These are discussed in detail in Section 5.

5 Discussion

This paper employs the CICIoT2023 dataset, comprising network traffic captured from multiple IoT environments. The dataset is imbalanced in nature, containing 97.7% of attack traffic and only 2.3% benign traffic. This level of imbalance is typical of IoT environments. Although model accuracy may appear to be around 97.7% due to majority-class bias, it is crucial to evaluate quantization approaches using the same bias

and see how their adversarial degradation varies. The extracted features have achieved 99.79% precision, 99.28% recall and ROC-AUC of 0.9964, indicating that the learning model is not merely learning to predict the majority class.

5.1 Quantization-Accuracy Trade-Off

Results demonstrate that INT8 quantization induces no statistically detectable accuracy loss on clean data while yielding substantial efficiency gains. On 40,000 clean test samples, McNemar's test found identical FP32 and INT8 predictions ($n_{01} = 0, n_{10} = 0, p = 1.0$), and all models achieved 0.9910 ± 0.0005 accuracy with 95% Wilson confidence intervals. Any possible accuracy difference is restricted to about ± 0.05 percentage points and thus is not meaningful for IDS deployment decision; we call this measurement-precision-limited equivalence rather than absolute accuracy identity. This stability may be attributed to batch normalization preceding quantization-sensitive layers, the low variance, low dimensionality, 41 feature normalized input space and INT8 input calibration using representative training samples to determine suitable scales and zero-points. However, this equivalence is statistical: smaller differences may be obscured by limited sample size; real-world distribution shifts may expose quantization sensitivity; and probability-sensitive tasks may degrade even when classification accuracy is preserved.

5.2 Energy Efficiency and Deployment Feasibility

The 92.7% energy reduction from FP32 to INT8 carries substantial implications for battery-operated IoT devices. However, energy model also reveals the unexpected fact that while computation energy is reduced by 44.8% from FP32 to INT8, memory energy is still dominating at 50.7% of the total energy for INT8 representations. This finding has profound implications for future optimization strategies. Model pruning eliminates redundant connections, reducing memory traffic by 30%–50% with minimal accuracy loss through removal of low-contribution weights. Weight clustering groups similar weights into a small number of clusters reducing the unique memory addresses stored in memory, hence improving the cache efficiency and potentially reducing the memory access energy by 20%–30%. ReLU produces many zeros in the activation values, the activations can be compressed by applying run-length encoding. Activation sparsity induced by ReLU enables compression achieving approximately 20%–40% memory bandwidth reduction. On solar-Powered IoT Nodes, INT8 requires only 0.0083 mJ per inference. Using one inference per minute, this corresponds to 0.0046 μW average power consumption. At this level, even a small 5 cm^2 solar panel operating at 10 μW under low illumination provides sufficient energy. This renders outdoor deployment for monitoring tasks feasible without battery replacement. We note that these results are derived from hardware-informed analytical estimation; physical validation on actual ARM Cortex-M4 hardware is planned as future work, with prior studies [27] reporting <2% deviation between estimation and measurement for TFLite INT8 models.

5.3 Adversarial Robustness: Transfer Attacks vs. Gradient Masking

The transfer attack methodology addresses a critical flaw in prior adversarial evaluation of quantized models. Quantization improves adversarial robustness by 3%–10% according to many studies, but most of these studies do not consider the impact of gradient masking, which occurs due to the non-differentiable nature of quantization operations and yields only illusionary robustness. The adversarial accuracy for FP32 and FP16 both saturates at 98.80% for $\varepsilon \geq 0.10$. This saturation suggests that the models have learned genuinely robust features. The slight decrease in adversarial accuracy of the model attacked by PGD means that adversarial samples generated by this method are not highly transferred to our model architecture. Several factors may contribute to this behavior: the 41-dimensional feature space likely offers a limited adversarial

subspace compared to high-dimensional image data, batch normalization adaptively rescales adversarial perturbations through input-dependent normalization, and the model's high-confidence predictions leave little room for adversarial exploitation. Meanwhile, the monotonic decrease in INT8 robustness by 1.6% at ε = 0.30 verifies that our transfer attack method can indeed evade gradient masking. This demonstrates that INT8 exhibits genuine rather than obfuscated robustness. Collectively, results demonstrate that INT8 maintains 97.50% accuracy under practical attacks, sufficient for most IoT security applications. In fact, the minor loss in robustness is well-compensated by the tremendous gain in terms of latency reduction (12×), and energy reduction (92.7%).

5.4 Limitations and Future Work

Several limitations of this study warrant acknowledgment. First, the use of a 200,000-sample subset, while statistically justified and computationally necessary, means the model has not been evaluated on the full CICIoT2023 dataset. Although our post-hoc sensitivity analysis showed minimal accuracy deviation (0.02%), future work should validate on the complete dataset or across multiple IoT benchmark datasets (e.g., UNSW-NB15, Bot-IoT, ToN-IoT) to confirm generalization. Second, our hardware-informed analytical estimation, while based on ARM Cortex-M4 specifications, does not substitute for on-device physical measurements. Future work should implement the INT8 model on actual hardware (e.g., STM32F429 Discovery board) to measure real-world latency and energy consumption. Third, our adversarial evaluation used transfer attacks from an FP32 source model to the INT8 target model; while this avoids gradient masking, it does not guarantee that a dedicated attacker cannot craft more powerful attacks specifically targeting the quantized model. Future work should explore quantization-aware adversarial training to recover the 1.6% robustness gap observed under PGD attacks at ε = 0.30.

6 Conclusion

This study demonstrates that aggressive INT8 quantization is highly suitable for constructing robust intrusion detection systems. The proposed solution achieves a distinctive metric set on the CIC-IoT2023 dataset for quantized models: 99.10% detection accuracy (matching the FP32 baseline), 12.0× speedup (to 0.083 ms), 92.7% energy reduction (to 0.0083 mJ per inference), and a memory footprint of 25.72 KB. By compressing the model to an established quantization of a compact 25.72 KB size, supported by batch normalization and careful calibration, incurs zero accuracy loss on clean data. In addition, the model demonstrated robustness to transfer attacks, where it sustained an accuracy of 97.50% at an adversarial perturbation level of $\epsilon = 0.30$. This demonstrates that the model is indeed geometrically robust and thus highly appropriate for IoT security, where the value of real-time and ultra-low power processing generally outweighs a small loss in robustness. Component-level energy profiling additionally reveals an energy inversion in quantized edge AI: memory access (50.7%) has superseded computation as the primary energy bottleneck. Future research will concentrate on two principal directions. First, investigation of sub-byte quantization (INT4) and binary neural networks to directly mitigate the memory fetch dominance; if accuracy can be preserved, this could potentially double battery life again. Second, to implement adversarial training directly on the quantized model to recover the 1.6% robustness loss observed under heavy attack. By integrating these algorithmic advancements with hardware constraints, we aim to establish a new standard for autonomous, perpetually operating edge security.

Acknowledgement: The authors would like to thank all individuals and institutions that contributed to this research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Saeed Ullah contributed to conceptualization, methodology, formal analysis, investigation, and writing—original draft preparation. Junsheng Wu contributed to conceptualization, resources, supervision, funding acquisition, and writing—review and editing. Mian Muhammad Kamal contributed to conceptualization, resources, supervision, funding acquisition, and writing—review and editing. Mohammed K. Alzaylaee contributed to methodology, formal analysis, and data curation. Heba G. Mohamed contributed to methodology, investigation, and writing—review and editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analysed during this study are included with in the manuscript itself.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rathore S, Bhandari A, Maini R. A multi-dimensional study of IoT DDoS in smart environments: SDN integration, taxonomy, security gaps, and emerging defenses. Comput Sci Rev. 2026;60(5):100903. doi:10.1016/j.cosrev.2026.100903.
2. Xu F, Yang HC, Alouini MS. Energy consumption minimization for data collection from wirelessly-powered IoT sensors: session-specific optimal design with DRL. IEEE Sens J. 2022;22(20):19886–96. doi:10.1109/JSEN.2022.3205017.
3. Statista. Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034. [cited 2026 Jan 1]. Available from: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.
4. Deivakani M, Sheela MS, Priyadarsini K, Farhaoui Y. An intelligent security mechanism in mobile Ad-Hoc networks using precision probability genetic algorithms (PPGA) and deep learning technique (Stacked LSTM). Sustain Comput Inform Syst. 2024;43(3):101021. doi:10.1016/j.suscom.2024.101021.
5. Canavese D, Mannella L, Regano L, Basile C. Security at the edge for resource-limited IoT devices. Sensors. 2024;24(2):590. doi:10.3390/s24020590.
6. Mansoor K, Afzal M, Iqbal W, Abbas Y. Securing the future: exploring post-quantum cryptography for authentication and user privacy in IoT devices. Cluster Comput. 2024;28(2):93. doi:10.1007/s10586-024-04799-4.
7. Walling S, Lodh S. An extensive review of machine learning and deep learning techniques on network intrusion detection for IoT. Trans Emerging Tel Tech. 2025;36(2):e70064. doi:10.1002/ett.70064.
8. Hnamte V, Ahmad Najar A, Nhung-Nguyen H, Hussain J, Sugali MN. DDoS attack detection and mitigation using deep neural network in SDN environment. Comput Secur. 2024;138(21):103661. doi:10.1016/j.cose.2023.103661.
9. Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.
10. Zhou Z, Shojafar M, Alazab M, Abawajy J, Li F. AFED-EF: an energy-efficient VM allocation algorithm for IoT applications in a cloud data center. IEEE Trans Green Commun Netw. 2021;5(2):658–69. doi:10.1109/tgcn.2021.3067309.
11. Alkasassbeh M, Omoush EH, Almseidin M, Aldweesh A. A self-adaptive intrusion detection system for zero-day attacks using deep Q-networks. IEEE Access. 2025;13:174280–96. doi:10.1109/ACCESS.2025.3617792.
12. Li X, Gong X, Wang D, Zhang J, Baker T, Zhou J, et al. ABM-SpConv-SIMD: accelerating convolutional neural network inference for industrial IoT applications on edge devices. IEEE Trans Netw Sci Eng. 2023;10(5):3071–85. doi:10.1109/TNSE.2022.3154412.
13. Sefati SS, Arasteh B, Halunga S, Fratu O. A comprehensive survey of cybersecurity techniques based on quality of service (QoS) on the Internet of Things (IoT). Cluster Comput. 2025;28(12):792. doi:10.1007/s10586-025-05449-z.
14. Banerjee R. Hands-on TinyML: harness the power of machine learning on the edge devices. New Delhi, India: BPB Publications; 2023.

15. Banait SSR, Anish CM, Sen M, Mondal D, Dhabliya D, Kumar JRR. Advancements in defense mechanisms against adversarial attacks in computer vision. In: Proceedings of the 6th International Conference on Information Management & Machine Intelligence. New York, NY, USA: The Association for Computing Machinery (ACM); 2024. p. 1–10. doi:10.1145/3745812.3745886.
16. Patel ND, Rao VS, Singh A. QDNN-IDS: quantized deep neural network based computational strategy for intrusion detection in IoT. In: Proceedings of the 2024 IEEE Silchar Subsection Conference (SILCON 2024); 2024 Nov 15–17. Agartala, India. p. 1–7. doi:10.1109/silcon63976.2024.10910857.
17. Selvaraj R, Kuthadi VM, Baskar DS, Acevedo R. Tiny ML-enabled energy-efficient intrusion detection system for sustainable IoT security in green cybersecurity ecosystems. JISIS. 2025;15(3):602–25. doi:10.58346/jisis.2025.i3.041.
18. Qi Z, Gu H, Chao Z, Dong Z, Wang H, Gershome AG. RVH-cover: a resultant-vector and heap-based WSN coverage optimization algorithm. IEEE Internet Things J. 2026;13(10):20811–21. doi:10.1109/jiot.2026.3664115.
19. STM32F429/439. Arm Cortex-M4 high performance microcontrollers (MCU). STMicroelectronics. [cited 2026 Feb 8]. Available from: https://www.st.com/en/microcontrollers-microprocessors/stm32f429-439.html.
20. Bommana SR, Veeramachaneni S, Ahmed SE, Srinivas MB. Addressing adversarial attacks in IoT using deep learning AI models. IEEE Access. 2025;13:50437–49. doi:10.1109/ACCESS.2025.3552529.
21. Anthi E, Williams L, Javed A, Burnap P. Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks. Comput Secur. 2021;108(7):102352. doi:10.1016/j.cose.2021.102352.
22. Sánchez Sánchez PM, Huertas Celdrán A, Bovet G, Martínez Pérez G. Adversarial attacks and defenses on ML- and hardware-based IoT device fingerprinting and identification. Future Gener Comput Syst. 2024;152(12):30–42. doi:10.1016/j.future.2023.10.011.
23. Gaber T, Ali T, Nicho M, Torky M. Robust attacks detection model for Internet of flying things based on generative adversarial network (GAN) and adversarial training. IEEE Internet Things J. 2025;12(13):23961–74. doi:10.1109/JIOT.2025.3555202.
24. Yuan X, Han S, Huang W, Ye H, Kong X, Zhang F. A simple framework to enhance the adversarial robustness of deep learning-based intrusion detection system. Comput Secur. 2024;137(1):103644. doi:10.1016/j.cose.2023.103644.
25. Kumar VS, Mohan Raj KR, Gopalakrishnan S, Vennila G, Dhinakaran D, Kavitha P. Adaptive distributed honeypot detection network for enhanced cybersecurity against DoS and DDoS attacks. Results Eng. 2025;26(3):105521. doi:10.1016/j.rineng.2025.105521.
26. Wang D, Du L, Li Z, Cheng N, Si J, Shen W. WPMNet: weight pruning and parameter quantization assisted multi-scale lightweight network for interference recognition. IEEE Trans Cogn Commun Netw. 2026;12:1313–26. doi:10.1109/tccn.2025.3614368.
27. Deng Z, Torim A, Ben Yahia S, Bahsi H. Generative AI in intrusion detection systems for Internet of Things: a systematic literature review. IEEE Open J Commun Soc. 2025;6(4):4689–717. doi:10.1109/OJCOMS.2025.3573194.
28. Sai S, Kanadia M, Chamola V. Empowering IoT with generative AI: applications, case studies, and limitations. IEEE Internet Things Mag. 2024;7(3):38–43. doi:10.1109/IOTM.001.2300246.
29. Mallikarjun A, Patro P. Hybrid data balancing with MLP probabilities-based categorical boosting model for robust intrusion detection system in IoT environment. Syst Soft Comput. 2026;8(16):200443. doi:10.1016/j.sasc.2026.200443.
30. Chen X, Wang P, Yang Y, Liu M. Resource-constraint deep forest-based intrusion detection method in Internet of Things for consumer electronic. IEEE Trans Consum Electron. 2024;70(2):4976–87. doi:10.1109/TCE.2024.3373126.
31. Nazir A, He J, Zhu N, Qureshi SS, Qureshi SU, Ullah F, et al. A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. Ain Shams Eng J. 2024;15(7):102777. doi:10.1016/j.asej.2024.102777.
32. Azimjonov J, Kim T. Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors. Comput Secur. 2024;137:103598. doi:10.1016/j.cose.2023.103598.
33. Kumar S, Keshri AK. An effective DDoS attack mitigation strategy for IoT using an optimization-based adaptive security model. Knowl Based Syst. 2024;299(2):112052. doi:10.1016/j.knosys.2024.112052.

34. Mallidi SKR, Ramisetty RR. Optimizing intrusion detection for IoT: a systematic review of machine learning and deep learning approaches with feature selection and data balancing. WIREs Data Mining Knowl Discov. 2025;15(2):e70008. doi:10.1002/widm.70008.
35. Zhou Z, Yin R, Wang W, Ni Q, Zarakovitis CC. Wireless deep mutual learning: challenges and opportunities. IEEE Commun Mag. 2026;64(6):88–94. doi:10.1109/mcom.001.2500728.
36. Abu Bakar R, De Marinis L, Cugini F, Paolucci F. FTG-Net-E: a hierarchical ensemble graph neural network for DDoS attack detection. Comput Netw. 2024;250(2):110508. doi:10.1016/j.comnet.2024.110508.
37. Ragunthar T, Christila SA, Jyoshna B, Meenachi L, Das S, Patro P. Machine learning-driven intrusion detection systems for anomaly detection in vehicular ad-hoc networks. Trans Emerging Tel Tech. 2026;37(2):e70363. doi:10.1002/ett.70363.
38. Xu H, Fang L, Dong J, Shi J. An efficient vehicular network anomaly detection framework based on encoder and dynamic threshold adjustment. Peer Peer Netw Appl. 2025;18(5):265. doi:10.1007/s12083-025-02093-7.
39. Merlino V, Allegra D. Energy-based approach for attack detection in IoT devices: a survey. Internet Things. 2024;27(4):101306. doi:10.1016/j.iot.2024.101306.
40. Syed S, Jayalakshmi S, Kumar Vankayalapati R, Mandala G, Yadav OP, Yadav AK. A robust and scalable deep learning framework for real-time IoT intrusion detection with adaptive energy efficiency and adversarial resilience. In: Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024); 2024 Oct 23–24; Tamilnadu, India. doi:10.2139/ssrn.5077791.
41. Xu H, Fang L. A stacking ensemble learning-based intrusion detection system for Internet of vehicles. Internet Technol Lett. 2025;8(5):e70046. doi:10.1002/itl2.70046.