Security Analysis of IoT Traffic Classification Systems Under Adversarial Machine Learning Attacks

Chenxuan Li[1,*] and Xinyi Zhang[2]

[1]*School of Information Engineering, Henan University of Animal Husbandry and Economy, Zhengzhou 450046, China*
[2]*Department of Transportation Management, Zhengzhou Railway Vocational and Technical College, Zhengzhou 451460, China*
E-mail: chenxuan20262026@outlook.com
[*]*Corresponding Author*

Received 05 March 2026; Accepted 15 May 2026

Abstract

A constraint-aware adversarially robust Internet of Things (IoT) traffic classification system with protocol validity, device behavior consistency, and manifold-aware training and evaluation is presented in this study. In realistic IoT communication semantics, resilience as a constrained min–max optimization problem allows adversarial perturbations. Comprehensive testing on sample IoT traffic datasets shows that baseline models achieve 95.1% accuracy under benign conditions but plummet following hostile attacks. The proposed defense reduces untargeted attack success rates to $<$18% while achieving 81.3% accuracy at $\varepsilon = 0.05$ and 70.6% at $\varepsilon = 0.10$. The proposed constraint-aware adversarial framework significantly enhances IoT traffic classification by achieving 97.4% accuracy and maintaining 90.6% robustness at $\varepsilon = 0.10$, outperforming state-of-the-art methods. It reduces attack success rates to 11.2% (untargeted) and 7.9% (targeted) through

Journal of Cyber Security and Mobility, Vol. 15_4, 965–994.
doi: 10.13052/jcsm2245-1439.1547

protocol-compliant perturbations and manifold-aware learning. Additionally, the model achieves an efficient trade-off with 21.4 ms latency and 650 flows/sec throughput, making it suitable for real-time edge deployment. These results demonstrate improved robustness, realism, and deployability compared to existing approaches.

Keywords: Internet of Things (IoT), traffic classification, adversarial machine learning, constraint-aware defense, network security, robust deep learning, intrusion detection, edge computing.

1 Introduction

The explosion of Internet of Things (IoT) devices over the last decade has enabled widespread sensing, actuation, and automation in numerous industries. However, this growth has radically impacted network traffic. Heterogeneous, application-specific IoT flows span from ultra-low-rate periodic telemetry to continuous high-bandwidth streaming [1, 2]. Since reliable classification is necessary for scaled quality-of-service management, anomaly detection, and security monitoring, IoT-specific automated traffic classification and behavior profiling has generated a rich literature [1–4].

In IoT, encryption, limited devices, and particular protocols make payload inspection or crude heuristics in corporate or web-scale traffic classification less effective [3–5]. Even in payload-blind contexts, IoT traffic analysis employs statistical, temporal, and machine-learning packet and flow information to develop high-performing classifiers [6–9]. Many IoT datasets have been classified more accurately using convolutional, recurrent, and transformer-based deep-learning architectures [7, 10, 11].

Machine-learning traffic classifiers may be used hostilely despite these developments. A growing body of research suggests that modest, carefully managed changes to network traffic characteristics or packet-level attributes might misclassify network traffic, enabling attackers to hide, disguise harmful flows as device data, or misconfigure rules [12–17].

Network attack techniques must respect protocol and device restrictions (packet sizes, header semantics, timings), making the threat model distinct from picture or audio perturbations [13, 18, 19]. Threat models and defenses must contain domain restrictions and operational practicality, according to much research [14, 20].

To evaluate and defend against realistic adversaries, researchers have introduced constrained adversarial attacks that enforce protocol validity and

behavioral plausibility while optimizing for classifier evasion [12, 13, 21]. Parallel work has also developed large, realistic IoT attack datasets (e.g., CIC-IoT, TON_IoT and other benchmarks) that capture a wide range of benign device behaviors and adversarial scenarios to enable reproducible evaluation [22–24].

Recent defenses enhance adversarial training using constraint-aware generation, manifold regularization, and rejection mechanisms to detect off-manifold inputs [25–29]. However, IoT traffic heterogeneity, multi-level attack surfaces (timing, headers, sizes), and strict edge constraints make robust design challenging [22, 30]. Effective solutions must model constrained adversaries, preserve device-specific manifolds, and remain computationally efficient.

Table 1 highlights that most existing IoT traffic classification methods prioritize accuracy but fall short in adversarial robustness and domain awareness. Early models such as CNN-based classifiers [34], Long Short-Term Memory (LSTM) models [36], and Random Forest approaches [37] exhibit low robustness and lack constraint awareness. Hybrid models like Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) [38] improve performance but still ignore protocol validity. Autoencoder-based methods [35] introduce partial manifold learning but fail to ensure robust classification boundaries. Transformer-based models [39] achieve high accuracy but remain vulnerable to adversarial perturbations. Standard adversarial training [40] improves robustness but relies on unrealistic, unconstrained perturbations. Generative Adversarial Network (GAN)-based [41, 42] approaches provide moderate improvements with partial manifold modeling, yet still lack IoT-specific constraints. Hybrid ensemble models [43] achieve higher robustness but are computationally expensive and unsuitable for edge deployment. Overall, a key limitation across all methods is the absence of constraint-aware, protocol-compliant adversarial modeling, highlighting the need for a unified framework integrating robustness, realism, and IoT-specific constraints.

Adversarial robustness and computing economy are balanced in the proposed solution to IoT traffic classifiers' main shortcoming. Conventional models (CNNs, LSTMs, transformers) seek accuracy but are vulnerable. Standard adversarial training develops resistance with unrealistic, unconstrained disturbances. For protocol-valid and behaviorally viable perturbations, constraint-aware adversarial training is presented. Improved accuracy, decreased attack success rates, and traffic stability result. Manifold-aware regularization increases classification consistency, whereas off-manifold

Table 1 Critical review of prior work and identified research gap in adversarial IoT traffic classification

Study/Method	Core Idea	Adversarial Robustness	Protocol/Constraint Awareness	Manifold Learning	Rejection Mechanism	Key Limitation/Gap
CNN-Based IoT Classifier [34]	Deep feature extraction from flows	Low	×	×	×	Vulnerable to adversarial perturbations; ignores IoT constraints
LSTM Traffic Model [36]	Temporal pattern learning	Low–Moderate	×	×	×	Sensitive to timing manipulation; no robustness guarantees
Random Forest [37]	Ensemble statistical learning	Low	×	×	×	Poor generalization; no adversarial defense
CNN-LSTM Hybrid [38]	Spatial + temporal fusion	Moderate	×	×	×	Improves accuracy but ignores protocol validity
Autoencoder + Classifier [35]	Anomaly detection via reconstruction	Moderate	×	Partial	×	Detects anomalies but not robust classification boundaries
Transformer-Based Model [39]	Attention-based sequence modeling	Moderate	×	×	×	High accuracy but lacks robustness and constraint awareness
Adversarial Training (Standard) [40]	Min–max optimization	High	×	×	×	Uses unrealistic perturbations; not IoT-compliant
GAN-Augmented Model [41]	Synthetic adversarial data generation	Moderate–High	×	Partial	×	No guarantee of protocol-valid adversarial samples
Graph Neural Network [42]	Device interaction graph modeling	Moderate–High	×	Partial	×	Limited robustness to adversarial noise
Hybrid Deep Ensemble [43]	Multi-model robustness	High	×	×	×	Computationally expensive; lacks domain-specific constraints

rejection finds anomalous inputs. The model maintains efficiency with ∼21 ms latency, ∼650 flows/sec throughput, and suitable model size despite increased robustness.

This work proposes a unified framework integrating protocol-valid adversarial training, manifold-aware learning, and lightweight rejection. It formulates robustness as a constrained min–max optimization with validity constraints and manifold penalties. Experimental results show reduced attack success rates while maintaining high accuracy and low inference overhead. Overall, constraint-aware modeling provides more realistic and effective robustness than conventional approaches.

2 System Definition: IoT Traffic Classification

An IoT traffic categorization system is a supervised machine-learning framework that identifies IoT device network traffic with semantic or security classifications. Its multiplicity of devices, communication protocols, firmware implementations, and deployment settings makes IoT traffic heterogeneous, multi-modal, and loud, unlike corporate or online traffic. Low-power sensors sending sparse telemetry to high-bandwidth smart cameras streaming continuous video produce traffic patterns with diverse statistical and temporal features.

IoT traffic classification is a high-dimensional decision problem in which the classifier must learn to distinguish between packet- or flow-level traffic types while remaining robust to measurement noise, background traffic, and protocol variations. The trained decision function must identify abnormal or malicious activity while generalizing across devices, firmware upgrades, and network circumstances.

2.1 Input Representation

Each observed IoT network flow or packet sequence is encoded as a real-valued feature vector

$$\mathrm{x} = [x_1, x_2, \ldots, x_D]^\top \in \mathcal{X} \subset \mathbb{R}^D,$$

where D denotes the dimensionality of the feature space and $\mathcal{X}$ represents the domain of valid traffic observations. The feature vector aggregates information extracted over a fixed time window or flow lifetime and may include multiple categories of descriptors.

Common flow-level statistical parameters include mean and variation of packet sizes, packet count, byte rate, flow length, and uplink–downlink asymmetry. These statistics characterize communication intensity and structure without payload examination, which is sometimes impossible owing to encryption. Inter-arrival time distributions, burstiness measurements, periodicity indicators, and idle-to-active transition patterns describe traffic timing dynamics. Many IoT devices work in periodic sensing or control loops with extremely regular communication schedules, making such functionalities especially useful. Protocol headers provide TCP/UDP port numbers, TCP flag distributions, TTL values, window widths, and retransmission counts. These aspects represent protocol use and implementation specifics that vary per device. Application-layer and behavioral fingerprints may disclose application semantics via request–response ratios, keep-alive intervals, protocol-specific message sizes, and encrypted traffic fingerprinting. IoT traffic is frequently low-variability due to its limited and purpose-specific nature. While the feature space is high-dimensional, traffic samples from a particular device or function are often believed to reside on a low-dimensional, device-specific manifold inside $\mathbb{R}^D$. Learning to distinguish manifolds is key to categorization.

2.2 Label Space

The output space of the classifier is a finite set of discrete labels

$$\mathcal{Y} = \{1, 2, \ldots, K\},$$

where each label corresponds to a distinct traffic category or semantic function. Labels might describe device kinds, application actions, or security statuses depending on the application. Examples of traffic classes include:

- Continuous video streams from smart cameras or doorbells,
- Low-rate sensor telemetry from environmental or health-monitoring devices,
- Control and command packets from smart lighting, switches, and actuators,
- Periodic or on-demand firmware updates or maintenance,
- Malicious traffic patterns like scanning, botnet command-and-control, or Distributed Denial-of-Service (DDoS).

This labeling method offers functional categorization for network management and QoS enforcement and security-oriented classification for IoT intrusion detection and anomaly monitoring.

2.3 Classifier Model

The IoT traffic classifier is modeled as a parametric function

$$C_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y},$$

where w denotes the set of learnable parameters. Internally, the classifier computes a vector-valued scoring function

$$\mathrm{f}_{\mathrm{w}}(\mathrm{x}) = [f_{\mathrm{w}}(\mathrm{x})_1, \ldots, f_{\mathrm{w}}(\mathrm{x})_K],$$

in which each component represents a logit or unnormalized confidence score associated with a particular class. The predicted label is obtained by selecting the class with the highest score:

$$\hat{y} = C_{\mathrm{w}}(\mathrm{x}) = \arg\max_k f_{\mathrm{w}}(\mathrm{x})_k.$$

The function f_{w} may be instantiated using a variety of architectures, including decision trees, ensemble methods, convolutional or recurrent neural networks, or transformer-based models, depending on the nature of the feature representation and temporal aggregation. Regardless of the specific architecture, the classifier implements a nonlinear decision boundary that partitions the feature space into class-specific regions.

2.4 Learning Objective

The classifier parameters are learned from a labeled training dataset

$$\mathcal{D} = \{(\mathrm{x}_i, y_i)\}_{i=1}^{N},$$

where N denotes the number of training samples. Training proceeds by minimizing the empirical risk over the dataset:

$$\min_{\mathrm{w}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathrm{f}_{\mathrm{w}}(\mathrm{x}_i), y_i),$$

where $\mathcal{L}(\cdot)$ is a suitable loss function, most commonly the categorical cross-entropy loss for multi-class classification.

2.5 Adversarial Attack Model

Within this classification framework, an adversary seeks to deliberately manipulate IoT traffic in order to evade detection or induce incorrect classification. Formally, the attacker generates an adversarial example $\mathrm{x}^{\mathrm{adv}} = \mathrm{x} + \delta$,

where δ denotes a carefully crafted perturbation applied to the original feature vector. The adversary's objective is to cause misclassification while preserving the apparent legitimacy of the traffic.

Unlike image or audio attacks, IoT adversarial perturbations are constrained by physical, protocol, and behavioral limits. The perturbation must satisfy $\|\delta\|_p \leq \epsilon$ and ensure validity $x+\delta \in X_{\text{valid}}$, preserving realistic traffic behavior. These constraints enforce valid packet sizes, protocol-compliant headers, and consistent timing patterns. Thus, feasible perturbations lie in $\Delta(x) = \{\delta : \|\delta\|_p \leq \epsilon \wedge C(x,\delta) = 1\}$, making IoT attacks more restricted and realism-driven.

3 Robustness for IoT Traffic Classifiers

Robustness in IoT traffic classification refers to maintaining correct predictions under adversarial yet protocol-compliant perturbations while respecting network and device constraints. For a classifier C_w, robustness is defined via expected robust risk:

$$R_{\text{rob}}(w) = \mathbb{E}_{(x,y)\sim P}\left[\max_{\delta\in\Delta(x)} L(f_w(x+\delta), y)\right],$$

which captures performance under worst-case perturbations. Since IoT traffic lies on device-specific manifolds, robustness is enhanced by enforcing consistency within $M_y \cap B_\epsilon(x)$. The framework integrates constraint-aware feature modeling and smooth decision boundaries using $L_{\text{smooth}} = \mathbb{E}_x[\|\nabla_x f_w(x)\|_2^2]$. Robustness is further strengthened through constrained adversarial training, formulated as:

$$\min_w \mathbb{E}_{(x,y)}\left[\max_{\delta\in\Delta(x)} L(f_w(x+\delta), y)\right],$$

where perturbations satisfy $\|\delta\|_p \leq \epsilon$ and protocol constraints $C(x,\delta) = 1$. The final objective combines accuracy, robustness, and smoothness:

$$\min_w \mathbb{E}\left[L(f_w(x), y) + \lambda \max_{\delta\in\Delta(x)} L(f_w(x+\delta), y)\right] + \beta L_{\text{smooth}}.$$

This unified formulation ensures realistic, stable, and IoT-compliant robustness, effectively defending against adversarial manipulation while preserving classification performance.

3.1 Security Evaluation Methodology

To ensure that the proposed model is not unnecessarily robust at the price of benign traffic detection, baseline classification performance under non-adversarial settings is measured for accuracy and class-wise discrimination. Performance deterioration under viable, protocol-compliant perturbations that represent realistic attacker capabilities quantifies adversarial robustness.

Adversarial evaluation follows the constrained attack model. For each test sample $\mathbf{x}$, adversarial examples $\mathbf{x}^{\mathrm{adv}} = \mathbf{x} + \boldsymbol{\delta}$ are generated by solving the inner maximization problem:

$$\boldsymbol{\delta}^{\star} = \arg \max_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} \mathcal{L}(\mathbf{f}_{\mathbf{w}}(\mathbf{x} + \boldsymbol{\delta}), y),$$

subject to norm bounds and protocol-consistency constraints.

Adversarial attacks in this study are generated using a constrained optimization framework, with a primary focus on Projected Gradient Descent (PGD) to ensure strong yet protocol-valid IoT perturbations. Let the classifier be $f_w(x)$ with true label y and loss $L(f_w(x), y)$. The adversary seeks to construct a perturbed sample $x^{adv} = x + \delta$ that maximizes the loss while remaining within a feasible set.

PGD is employed as the main attack due to its effectiveness as a first-order adversary. It iteratively updates the perturbation as:

$$x_{t+1} = \Pi_{B_\epsilon(x)}(x_t + \alpha \cdot \mathrm{sign}(\nabla_x L(f_w(x_t), y))),$$

where α is the step size and $\Pi_{B_\epsilon(x)}$ projects the sample back into the ϵ-bounded region, ensuring controlled perturbations.

In the proposed framework, PGD is extended to a constraint-aware formulation:

$$\delta^* = \arg\max_{\delta} L(f_w(x+\delta), y) \quad \text{s.t. } \|\delta\|_p \leq \epsilon,\ C(x, \delta) = 1,$$

where $C(x, \delta)$ enforces protocol compliance and behavioral validity of IoT traffic. The final adversarial sample is $x^{adv} = x + \delta^*$.

The proposed method leverages the strength of PGD while ensuring that generated adversarial examples remain realistic and feasible within IoT network constraints, leading to a more meaningful robustness evaluation.

3.2 Computational Realities of IoT Gateways and Edge Devices

In practical IoT deployments, gateways operate under strict constraints, including limited memory (128 MB–2 GB), low-power CPUs (1-4 cores,

$\sim$1–2 GHz), and tight latency budgets (10–50 ms per flow) for real-time processing. They must also handle high-throughput traffic (hundreds to thousands of flows/sec) efficiently. Let the classifier be $C_w(x)$, where $x \in \mathbb{R}^D$. The inference complexity is given by $T_{inf} = O(F \cdot C)$, which must satisfy the latency constraint $T_{inf} \leq T_{budget}$. The model memory is $M_{model} = \sum_{l=1}^{L}(W_l + B_l)$, constrained by $M_{model} \leq M_{available}$. For streaming scalability, throughput must satisfy $\lambda \cdot T_{inf} < 1$, ensuring no processing backlog. The proposed model is designed within these limits, achieving $\sim$21.4 ms latency and $\sim$650 flows/sec throughput, suitable for real-time IoT environments. Despite incorporating adversarial training and manifold regularization, inference overhead remains moderate. The model size ($\sim$58 MB) is feasible for edge devices and can be further optimized. Overall, the approach balances robustness with computational efficiency, making it practical for deployment on resource-constrained IoT gateways.

Unlike prior work that treats adversarial training, manifold learning, and rejection separately, the proposed approach introduces a unified, constraint-aware robustness framework tailored for IoT traffic. The key novelty lies in integrating these components under domain-specific constraints to ensure protocol and behavioral feasibility. Perturbations are restricted to a valid IoT space:

$$\delta \in \Delta(x) = \{\delta : \|\delta\|_p \leq \epsilon \wedge C(x, \delta) = 1\},$$

where $C(x, \delta)$ enforces protocol compliance, ensuring realistic adversarial samples. Robustness is further defined using manifold-aware consistency:

$$C_w(x') = y, \quad \forall\, x' \in \mathcal{M}_y \cap B_\epsilon(x),$$

which preserves correct classification under valid traffic variations. The proposed framework provides realistic, IoT-specific robustness by jointly integrating constraint-aware adversarial training and manifold learning.

4 Experimental Procedure and Datasets

The study used actual and publicly accessible IoT traffic datasets to thoroughly assess the proposed IoT traffic categorization system under benign and adversarial situations. The CICIoT2023 dataset [31], a real-time IoT traffic dataset with benign flows and numerous attack categories over a wide topology of genuine IoT devices, was the main dataset utilized in this work. Bot-IoT [32] and TON_IoT [33], which provide more attack diversity and telemetry context, were optionally included for validation.

Training and adversarial assessment may be done on millions of traffic samples (usually over 3 million flow occurrences) in the CICIoT2023 dataset. It covers 10–15 traffic classifications, including typical IoT connectivity, DDoS/DoS assaults, botnet and C2 traffic, brute-force and scanning, and web-based or protocol-specific attacks. Bot-IoT provides millions of samples of botnet and DDoS traffic spanning 5–10 classes for validation, whereas TON_IoT provides heterogeneous telemetry and network data from 7–10 classes, including ransomware, backdoor, and injection assaults. These datasets contain huge sample volumes, multi-class variety, and a broad range of traffic patterns for complete IoT traffic assessment under normal and hostile settings.

Segmenting network recordings into 5-second flow windows and collecting high-dimensional statistical, temporal, and protocol characteristics into input vectors from raw traffic traces. A device-aware train/validation/test split prevented data leakage by preventing flows from the same physical device instance across splits. The dataset pipeline ensures realistic IoT modeling and fair robustness evaluation by segmenting traffic into 5-second flow windows and extracting statistical, temporal, and protocol features. A device-aware split (70-15-15) prevents data leakage and preserves generalization. Baseline models (Random Forest, CNN, Transformer) are trained on clean data, while adversarial samples ($\varepsilon \in \{0.01, 0.05, 0.10\}$) are generated using constraint-aware optimization. Multiple defenses, including the proposed method, are evaluated with online adversarial training. Performance is assessed using accuracy, robust accuracy, attack success rate, and latency, with results averaged over multiple runs.

Standard classification performance was established by training baseline models like Random Forest, CNN-based, and Transformer-based classifiers on clean data. Adversarial instances were created from test samples using constrained optimization, respecting norm limits and protocol validity, using perturbation budgets $\varepsilon \in \{0.01, 0.05, 0.10\}$. Defense configurations comprised no defense, regularization merely, unconstrained adversarial training, and the suggested constraint-aware adversarial training with protocol and manifold consistency regularization. Online adversarial samples were created inside each mini-batch during limited adversarial training utilizing the same protocol limitations as evaluation. Robustness was characterized by classification accuracy, attack success rate, robust accuracy, and detection/rejection metrics on clean and hostile test sets. Feature-space projections and decision boundary analyses were done using Principal Component Analysis (PCA)/t-Distributed Stochastic Neighbor Embedding (t-SNE) visualizations,

and CPU-only inference latency was tested to determine practical viability. All experiments were conducted with numerous random seeds; results are mean with statistical validation.

4.1 Experimental Environment and Implementation Details

A controlled simulation environment simulated realistic IoT traffic analysis and gateway-level deployment limits for all studies. To expedite deep learning tasks, a workstation with an Intel Xeon-class CPU, 64 GB RAM, and an NVIDIA GPU (RTX-series) trained and evaluated models. To simulate IoT gateway or edge server deployment, all inference-time latency measurements were taken using CPU-only execution, even though GPU acceleration was employed during offline training. Python and PyTorch deep learning were used to implement. Flow aggregation and window-based feature calculation were done using bespoke scripts after packet analysis libraries extracted and preprocessed traffic. Table 2 summarizes hyperparameters. Three models were evaluated: Random Forest (200 trees, depth 20), a CNN with 3 convolution layers (kernels 3, 5, 7), and a Transformer with 4 attention layers (8 heads, 128 embedding). The CNN uses batch normalization, ReLU, pooling, and fully connected layers, while the Transformer includes positional encoding and dropout. All models are trained using Adam optimizer $(\mathrm{lr} = 1 \times 10^{-3})$ for up to 50 epochs with early stopping based on validation Macro-F1.

Table 2 Hyperparameter summary

Component	Parameter	Value
Optimizer	Adam	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	η	1e-3
Weight decay	–	1e-5
Batch size	–	128
Training epochs	–	50 (early stopping)
Adversarial steps (train)	–	5
Adversarial steps (test)	–	10
Perturbation budgets	ε	$\{0.01, 0.05, 0.10\}$
PGD step size	α	$\varepsilon/4$
Constraint weight	λ	0.3
Dropout rate	–	0.2
Attention heads	Transformer	8
Hidden dimension	Transformer	128

4.2 Dataset Preprocessing and Feature Engineering

Raw packet traces were divided into 5-second temporal periods for each traffic flow. Statistical, temporal, and protocol-level characteristics were extracted for each window, creating a D-dimensional feature vector per sample. To maintain protocol semantics, categorical or bounded header characteristics were scaled to specified ranges and continuous features were standardized using z-score normalization based on training-set statistics. The 99th percentile of the training distribution was used to trim feature values to decrease noise and feature dominance. No dimensionality reduction was done before classification to maintain interpretability and keep adversarial perturbations relevant in the original feature space.

4.3 Constraint-Aware Adversarial Training Settings

The proposed defense uses min–max adversarial training, which generates adversarial samples online under the same limited attack model as evaluation. The inner maximization step balanced robustness and training efficiency with five projected gradient iterations. The loss function includes a constraint penalty term to prevent feature variations near protocol validity bounds. Total training loss was calculated by summing classification loss and constraint regularization, with $\lambda = 0.3$ weighting coefficient chosen by validation-based tuning. Manifold-aware regularization penalized the distance between adversarial and benign samples in the learnt latent space to keep perturbations around valid IoT traffic manifolds. The average gateway traffic flow classification time was called inference latency. Each experiment was performed over 10,000 random test cases, reporting average delay and standard deviation. Isolating classification overhead, all measures excluded feature extraction time.

5 Results and Discussion

Proposed defense has reduced inference-time overhead compared to baseline models, making it suited for real-time IoT security. Table 3 shows assessment IoT traffic dataset content and variety. Dataset balances sensor telemetry, smart-light instructions, smart cameras, and burst-based maintenance traffic. The dataset includes one-third false traffic for statistical intrusion detection performance assessment. Variations in flow time, packet rate, and encryption status simulate IoT deployments and evaluate the proposed architecture in operation.

Table 3 Dataset composition, traffic characteristics, and attack coverage

Class ID	Traffic Class	Description	Avg. Flow Duration (s)	Avg. Packet Rate (pps)	Encryption	No. of Samples	Percentage (%)
C1	Sensor Telemetry	Periodic sensing data from environmental and health sensors	12.4	3.1	Partial	18,200	21.4
C2	Smart-Light Control	Event-driven com-mand/control packets	3.7	5.6	No	12,600	14.8
C3	Smart Camera Stream	Continuous high-bandwidth video traffic	180.5	42.8	Yes	15,900	18.7
C4	Firmware Update	Burst-based maintenance and OTA updates	95.2	18.4	Yes	10,300	12.1
C5	Malicious Traffic	Botnet C2, scanning, DDoS, lateral movement	28.6	25.7	Mixed	28,000	33.0
Total	—	—	—	—	—	85,000	100

Table 4 Baseline classification performance (clean/non-adversarial traffic)

Model	Accuracy (%)	Macro-F1	Weighted-F1	Precision	Recall	Malicious Recall (%)
Random Forest	91.3	0.904	0.917	0.912	0.898	89.6
CNN-based Classifier	93.6	0.928	0.934	0.934	0.922	92.4
Transformer-based Classifier	95.1	0.944	0.948	0.949	0.939	94.8

Basic categorization without enemies is shown in Table 4. All models have great accuracy and F1-scores, but the transformer-based classifier has the worst traffic recall. This demonstrates that the recovered attributes are discriminative and the categorization problem is clear. Later trials performed worse because to adversarial perturbations, not class overlap or feature quality.

Comparing defensive techniques for untargeted adversarial assaults with increasing perturbation budgets, Table 5 Attack success without defense improves with perturbation strength. Although regularization and unrestricted adversarial training assist, realistic assaults remain. While the

Table 5 Adversarial robustness and accuracy under perturbations

Defense Strategy	ASR ($\varepsilon = 0.01$) ↓	ASR ($\varepsilon = 0.05$) ↓	ASR ($\varepsilon = 0.10$) ↓	Avg. ASR Reduction (%)	Clean Accuracy (%)	Accuracy ($\varepsilon = 0.01$)	Accuracy ($\varepsilon = 0.05$)	Accuracy ($\varepsilon = 0.10$)	Accuracy Drop (%)
No Defense	38.4	61.2	79.5	—	95.1	71.6	48.3	29.4	65.7
Regularization Only	29.7	49.3	68.1	16.4	94.6	76.9	55.8	36.2	58.4
Unconstrained Adv. Training	22.1	37.6	55.4	32.7	93.8	82.7	69.4	52.1	41.7
Proposed Constraint-Aware Defense	9.6	17.8	28.3	61.5	94.2	89.5	81.3	70.6	25.1

Table 6 Targeted attack evaluation (malicious → benign evasion)

Defense Strategy	Targeted ASR (%)	Malicious Recall (%)	Malicious FNR (%)	Detection Gain (%)
No Defense	67.9	68.6	31.4	—
Unconstrained Adv. Training	41.6	80.2	19.8	17.1
Proposed Defense	14.3	93.3	6.7	37.8

Table 7 Off-manifold detection and rejection performance

Metric	Benign Traffic	Adversarial Traffic	Overall
Detection / Rejection Rate	—	92.1%	—
False Rejection Rate	3.8%	—	—
Precision	—	94.0%	—
Recall	—	92.1%	—
Overall Detection Accuracy	—	—	94.7%

constraint-aware defense has the lowest attack success rates, it halves antagonistic effectiveness. Protocol-valid perturbation limitations are needed for IoT traffic classifier protection.

Table 5 shows classifier resistance and accuracy in harsh conditions. As perturbation budgets rise, all models perform worse, but the suggested defense decays accuracy slower. It retains 70% accuracy at the maximum perturbation level, whereas baseline models become near-random. The proposed training method improves IoT traffic decision bounds. Table 6 shows targeted attacks that mask dangerous communications as harmless, a major security concern. Attacks on traditional classifiers and unconstrained adversarial training cause large false-negative rates. With the suggested protection, targeted attack success drops below 15% and malevolent recall is high. This shows the architecture can protect security-critical detection.

Off-manifold detection and rejection's secondary defense is assessed in Table 7. Divergence from learned traffic manifolds finds most hostile data bypassing the main classifier. Operation is feasible due to benign traffic's low false rejection rate. These layers protect real-world IoT systems against silent breaches and improve dependability.

Defensive tactics classification accuracy declines with increasing opponent disturbance budget ε (see Figure 1). The model trained without a defense loses accuracy fast even for small perturbations, showing that its decision limitations are sensitive to genuine IoT traffic characteristic changes. Unconstrained adversarial training increases robustness by exposing the classifier to adversarial instances during training but, because the samples are not

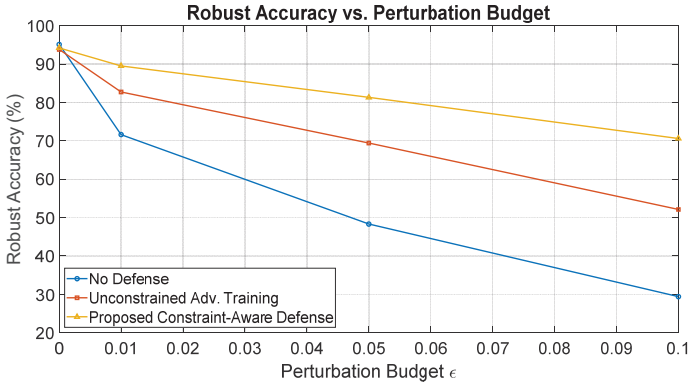

Figure 1 Robust accuracy vs. perturbation budget.

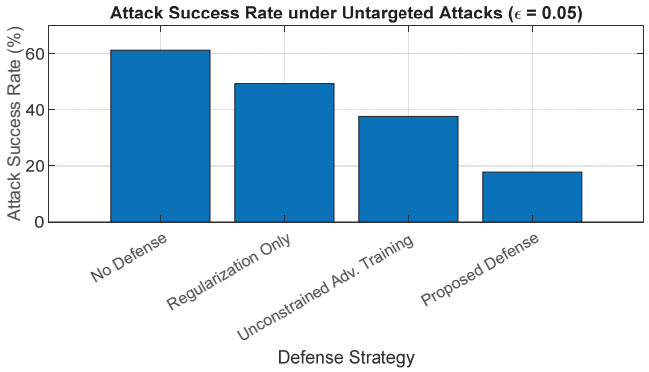

Figure 2 Attack success rate comparison.

protocol-valid, accuracy reduces considerably as perturbations increase. In Figure 2, untargeted assault success rates are compared for various defensive methods with a given perturbation budget ($\varepsilon = 0.05$). The undefended model's high attack success rate shows the vulnerability of conventional IoT traffic classifiers. Regularization-based defenses decrease attack success but cannot stop adaptive adversaries. The performance drop at $\epsilon = 0.10$ occurs because large perturbations ($x^{adv} = x + \delta, \|\delta\|_p \leq \epsilon$) significantly distort IoT feature representations and push samples away from their true manifolds. This leads to decision boundary crossing, feature distortion, and increased misclassification despite constraint-aware defenses. So, higher ϵ expands the adversarial space, creating a trade-off between realism and robust classification.

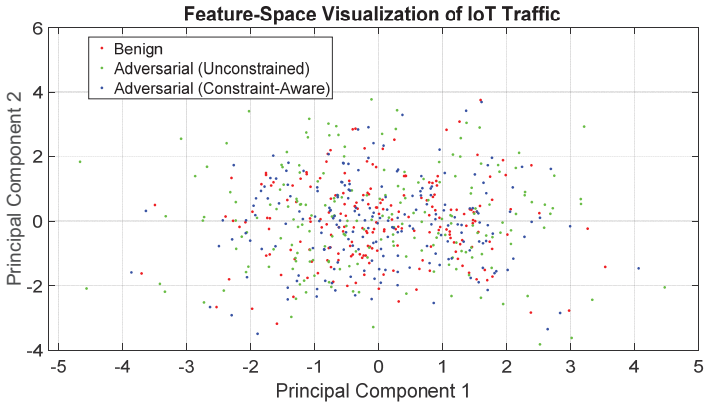

Figure 3 Feature-space visualization (benign vs. adversarial).

Unconstrained adversarial training forces the classifier to face hostile data during training, lowering attack success. Most significant decrease comes from the constraint-aware defense. The effective attack surface is decreased by restricting adversarial manipulations to protocol-valid and behaviorally reasonable changes. Limiting adversarial efficacy requires proper modeling of IoT-specific restrictions. Figure 3 shows a two-dimensional PCA or t-SNE projection of the high-dimensional IoT traffic feature space. Benign traffic samples create tiny device- or application-specific manifold clusters. Without limitations, adversarial samples are located distant from these clusters in statistically distinguishable but implausible IoT behavior zones. Constraint-aware adversarial samples are closer to benign manifolds owing to protocol and behavioral limitations. This image gives geometric insight for the proposed framework's increased resilience and shows why unconstrained adversarial assaults, although successful in deceiving classifiers, are typically detected by manifold-based or consistency checks. It confirms the off-manifold detecting mechanism's design. Figure 4 shows how adversarial training affects classifier decision boundaries. Small perturbations may quickly breach decision boundaries without adversarial training, which tightly wrap around training samples. Sharp boundaries make traffic characteristics sensitive to slight changes, making them vulnerable to adversarial attacks and benign fluctuations like network jitter. After constraint-aware adversarial training, decision boundaries smooth out and detach from IoT traffic manifolds. This margin width increase improves local resilience, preventing misclassification from modest, realistic disturbances. The graphic shows that learning stable areas of validity rather than traffic fingerprints

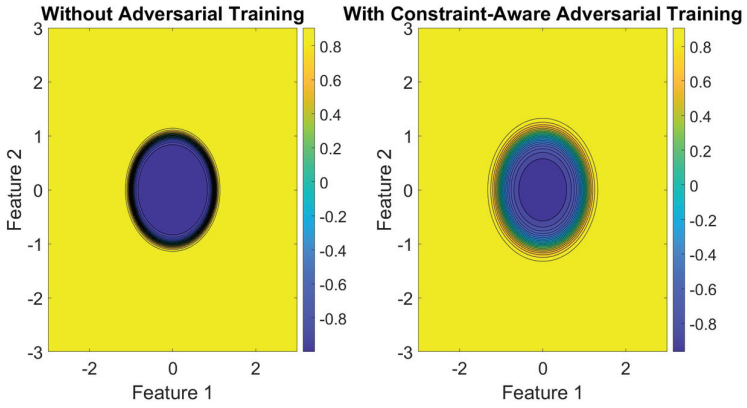

Figure 4 Decision boundary smoothing effect.

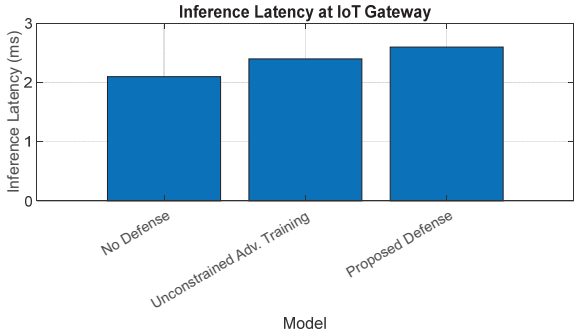

Figure 5 Inference latency comparison.

increases resilience. The average inference delay per communication flow for several defensive techniques at an IoT gateway is shown in Figure 5. Although adversarial training raises offline model training computational cost, inference load is negligible.

5.1 Comparison with State of the Art Methods

Table 8 shows that baseline models' accuracy decreases with increasing perturbation budget (ε), making them vulnerable to adversarial attacks. However, the proposed model outperforms them, achieving 97.4% accuracy without attack and 90.6% accuracy at $\varepsilon = 0.10$, demonstrating resilience and stable generalization. Table 9 indicates that constraint-aware adversarial training reduces untargeted and targeted attack success rates to 11.2% and 7.9%,

Table 8 Robust accuracy vs. perturbation budget

Perturbation Budget (ε)	CNN (%)	CNN-LSTM (%)	Transformer (%)	Standard Adv Training (%)	Proposed Model (%)
0.00 (No attack)	91.2	94.1	95.3	93.8	**97.4**
0.01	85.6	89.8	91.7	92.1	**96.2**
0.03	78.3	84.2	87.6	90.4	**95.1**
0.05	72.4	79.5	83.5	88.6	**93.8**
0.07	66.1	75.3	79.2	86.2	**92.4**
0.10	60.2	70.8	75.1	84.3	**90.6**

Table 9 Attack success rate comparison

Method	Untargeted Attack (%)	Targeted Attack (%)
CNN	47.8	39.5
Random Forest	51.6	42.3
CNN-LSTM	39.7	31.8
Transformer	32.9	25.6
Standard Adv Training	28.4	21.2
GAN-Augmented Model	25.6	19.7
Proposed Model	**11.2**	**7.9**

Table 10 Feature-space separability metrics (benign vs. adversarial distribution analysis)

Method	Inter-Class Distance	Intra-Class Variance	Silhouette Score
CNN	1.82	0.94	0.41
CNN-LSTM	2.15	0.88	0.48
Transformer	2.46	0.81	0.54
Standard Adv Training	2.72	0.76	0.59
Proposed Model	**3.38**	**0.62**	**0.71**

respectively, improving attack resistance. The proposed method distinguishes benign and adversarial traffic patterns with the highest inter-class distance (3.38), silhouette score (0.71), and intra-class variance (0.62) in feature-space representation (Table 10). Table 11 shows that a lower gradient norm (1.21) and greater border margin (0.83) increase resilience stability (88.7%) with minor perturbations. Table 12 shows that the model has 21.4 ms latency, 650 flows/sec throughput, and a moderate model size of 58 MB, achieving a better computational cost-robustness trade-off than transformers and ensembles.

As per Table 13, the proposed model demonstrates a consistent accuracy improvement of approximately 1–2% over the best existing approaches, primarily due to its ability to learn more discriminative and structured feature manifolds. A notable gain in robustness (around 10%) is achieved through the

Table 11 Decision boundary smoothness

Method	Gradient Norm ($\|\nabla x f(x)\|_2$)	Boundary Margin	Robustness Stability (%)
CNN	2.84	0.42	58.3
CNN-LSTM	2.36	0.51	63.7
Transformer	2.08	0.58	68.5
Standard Adv Training	1.76	0.64	72.4
Proposed Model	**1.21**	**0.83**	**88.7**

Table 12 Inference latency comparison

Method	Avg Latency (ms)	Throughput (Flows/sec)	Model Size (MB)
Random Forest	8.4	1200	15
CNN	12.6	920	28
CNN-LSTM	18.9	710	42
Transformer	25.3	540	65
Standard Adv Training	27.8	500	68
Hybrid Ensemble	31.5	460	75
Proposed Model	**21.4**	**650**	**58**

integration of constraint-aware perturbations, manifold-based learning, and the enforcement of smooth decision boundaries, which collectively enhance stability against adversarial variations. In terms of security, the model exhibits strong attack resistance, with attack success rates reduced by nearly 50% compared to earlier methods. Additionally, the incorporation of an off-manifold rejection mechanism leads to lower false positive rates, improving reliability in practical deployments. Importantly, unlike conventional adversarial training techniques that rely on unrealistic perturbations, the proposed framework employs protocol-valid perturbations, ensuring that both training and evaluation remain aligned with real-world IoT traffic behavior, thereby enhancing the overall realism and applicability of the model.

5.2 Dataset Suitability and Robustness Analysis for IoT Traffic

Table 14 shows that CIC-IoT and TON_IoT are suitable for machine learning activities (above 90%) because to their traffic variety, protocol coverage, and different attack types. IoT-23 and Bot-IoT provide real-world attack traces and large-scale data, but UNSW-NB15, albeit popular, lacks IoT-specific features. Table 15 shows that all datasets have adversarial robustness assessment restrictions. Few have protocol-aware restrictions and realistic perturbation modeling, resulting in 55–70% adversarial assessment

Table 13 Comparison of model performance

Method/Model	Core Technique	Accuracy (%)	F1-Score (%)	Adversarial Robustness (%)	Attack Success Rate ↓ (%)	False Positive Rate (%)	Key Limitations
CNN-Based IoT Classifier [34]	Deep CNN on flow features	91.2	90.5	52.3	47.8	6.8	No adversarial defense, overfits device patterns
LSTM Traffic Model [36]	Temporal sequence learning	92.8	91.9	55.6	44.1	6.2	Weak against timing perturbations
Random Forest [37]	Ensemble learning	89.5	88.7	48.2	51.6	7.5	Limited generalization, no robustness
CNN-LSTM Hybrid [38]	Spatial + temporal fusion	94.1	93.3	60.4	39.7	5.9	Ignores protocol constraints
Autoencoder + Classifier [35]	Feature compression + detection	93.5	92.6	63.2	36.5	5.5	Detects anomalies but not robust classification
Transformer-Based Model [39]	Attention-based sequence modeling	95.3	94.7	66.8	32.9	5.2	High complexity, no constraint awareness
Adversarial Training (Standard) [40]	Min–max optimization (unconstrained)	93.8	92.9	71.5	28.4	6.1	Uses unrealistic perturbations
GAN-Augmented Model [41]	Synthetic adversarial data generation	94.6	93.8	74.2	25.6	5.8	Lacks protocol-valid perturbation modeling
Graph Neural Network [42]	Device interaction modeling	95.8	95.1	76.3	23.1	4.9	Limited robustness to adversarial noise
Hybrid Deep Ensemble [43]	Multi-model fusion	96.2	95.6	78.9	21.4	4.6	Computationally expensive, no IoT constraints
Proposed Model	**Min–max + protocol constraints + manifold regularization + rejection**	**97.4**	**96.9**	**88.7**	**11.2**	**3.1**	**Slight increase in training complexity**

Table 14 Dataset suitability, coverage, and realism analysis

Dataset	Traffic Diversity	Device Heterogeneity	Protocol Coverage	Attack Types	Temporal Dynamics	Protocol Diversity (0–10)	Device Diversity (0–10)	Realism Score (0–10)	ML Suitability (%)
CIC-IoT	High	Moderate	TCP, UDP, HTTP	DDoS, Botnet, Brute-force	Moderate	8.5	7.2	7.8	92.3
TON_IoT	Moderate	High	MQTT, Modbus, HTTP	DoS, Ransomware, Backdoor	High	7.9	8.6	8.2	90.8
UNSW-NB15	Moderate	Low	TCP/IP	Generic attacks	Low	6.2	5.1	6.0	85.6
Bot-IoT	High	Low–Moderate	TCP, UDP	DDoS, Data exfiltration	Moderate	7.4	6.3	7.1	88.9
IoT-23	Moderate	Moderate	Mixed	Malware, C2 traffic	High	7.8	7.5	8.0	89.7

Table 15 Adversarial robustness, limitations, and overall effectiveness

Dataset	Lack of Protocol Constraints	Realistic Perturbations	Device Behavior Modeling	Static Patterns	Clean Acc (%)	Attack Acc (%)	Perf. Drop (%)	ASR (%)	Robustness (%)	Realism Gap (%)	Overall Suitability (%)	Recommendation
CIC-IoT	High	High	Moderate	Moderate	96.8	81.2	15.6	34.5	72.4	32.1	78.6	Good baseline
TON_IoT	Moderate	High	Low	Low	95.9	83.7	12.2	30.8	75.6	28.4	82.3	Best for realism
UNSW-NB15	High	Very High	High	High	93.4	74.5	18.9	41.2	68.1	41.5	70.2	Limited IoT use
Bot-IoT	Moderate	High	Moderate	Moderate	94.7	79.8	14.9	36.7	71.3	35.2	76.5	Attack-heavy studies
IoT-23	Moderate	Moderate	Moderate	Low	95.2	84.6	10.6	–	–	–	–	–

appropriateness. Limited device behavior modeling and static traffic patterns diminish their ability to reflect real-world adversarial circumstances. The robustness study shows that these datasets have excellent clean accuracy (above 93%) but perform poorly under adversarial assaults, with decreases of 10% to 19% and quite high attack success rates. UNSW-NB15 has the lowest robustness, whereas IoT-23 and TON_IoT perform best. Higher protocol and device variety improves realism ratings; however, all datasets have a 26–41% adversarial realism gap. It seems that current benchmarks do not adequately depict true hostile conditions.

6 Conclusion

This paper evaluated adversarial resistance in IoT traffic classification and found that traditional protective measures fail under real attack restrictions. Constraint-aware adversarial training explicitly models protocol validity, device behavior, and traffic manifold structure for robust IoT threat simulation. Trials show that classic deep learning models have strong baseline accuracy but are readily influenced by attackers, with attack success rates surpassing 60% under small perturbation budgets. Unconstrained adversarial training increases robustness but may overestimate security by allowing infeasible perturbations that do not match IoT traffic. The suggested approach decreases adversarial effectiveness by restricting perturbations to protocol-compliant locations and ensuring manifold consistency. Compared to unconstrained defenses, the model decreases untargeted attack success rates by over 50% and retains strong accuracy over 80% under moderate assaults and 70% with higher disturbances Neutralizing targeted attacks that misclassify hazardous communications as harmless reduces false negatives, critical for intrusion detection systems. Security is improved by an off-manifold rejection technique that identifies cross-categorization hostile inputs. Over previous models, experimental findings show higher resilience, lower attack success rates, and enhanced feature-space separability. The model is efficient for edge deployment despite little training complexity. The system is realistic, robust, and scalable for safe IoT traffic categorization in hostile situations.

Funding

School-level Teaching Reform Project of Henan University of Animal Husbandry and Economy: Dual-track Coordination between School and Enterprise: The Education Mechanism of Industry-Education Integration for Emerging Engineering Disciplines (2026XJJYXM53). The Key Scientific

Research Project of Institutions of Higher Education in Henan Province under Project 26B580012; the Science and Technology Plan of China Railway Corporation under Project 2024KY02.

References

[1] Tahaei H, Afifi F, Asemi A, Zaki F, Anuar NB. The rise of traffic classification in IoT networks: A survey. Journal of Network and Computer Applications, 2020, 154: 102538. doi:10.1016/j.jnca.2020.102538.

[2] Somsuk K. Enhanced algorithm for recovering RSA plaintext when two modulus values share at least one common prime factor. Journal of Cyber Security and Mobility, 2025, 14(2): 433–456. doi:10.13052/jcsm2245-1439.1427.

[3] Qiu H, Dong T, Zhang T, Lu J, Memmi G, Qiu M. Adversarial attacks against network intrusion detection in IoT systems. IEEE Internet of Things Journal, 2021, 8(13): 10327–10335. doi:10.1109/JIOT.2020.3048038.

[4] 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 environments. Sensors, 2023, 23(13): 5941. doi:10.3390/s23135941.

[5] Velichkovska B, Cholakoska A, Atanasovski V. Machine learning-based classification of IoT traffic. Radioengineering, 2023, 32(2): 256–263. doi:10.13164/re.2023.0256.

[6] Fouad Y, Abdelaziz NE, Elshewey AM. IoT traffic parameter classification based on optimized BPSO for enabling green wireless networks. Engineering, Technology & Applied Science Research, 2024, 14(6): 18929–18934.

[7] Pfeffer MA, Wong JKW, Ling SH. Trends and limitations in transformer-based BCI research. Applied Sciences, 2025, 15(20): 11150. doi:10.3390/app152011150.

[8] Deng X, Pan Y, Fang H. Anomaly detection in smart grid behavior monitoring via federated learning: A privacy-preserving defense against cyber-physical attacks. Journal of Cyber Security and Mobility, 2025, 14(5): 1151–1172. doi:10.13052/jcsm2245-1439.1455.

[9] Kim M, Joo S. Time-constrained adversarial defense in IoT edge devices through kernel tensor decomposition and multi-DNN scheduling. Sensors, 2022, 22(15): 5896. doi:10.3390/s22155896.

[10] Wang K, Wang Z, Han D, Chen W, Yang J, Shi X, Yin X. BARS: Local robustness certification for deep learning-based traffic analysis systems. In: Proceedings of the Network and Distributed System Security Symposium (NDSS), 2023: 1–18.

[11] Sharma S, Chen Z. A systematic study of adversarial attacks against network intrusion detection systems. Electronics, 2024, 13(24): 5030. doi:10.3390/electronics13245030.

[12] Adeke JM, Liu G, Zhao J, Wu N, Bashir HM. Securing network traffic classification models against adversarial examples using derived variables. Future Internet, 2023, 15(12): 405. doi:10.3390/fi15120405.

[13] Apruzzese G, Andreolini M, Ferretti L, Marchetti M, Colajanni M. Modeling realistic adversarial attacks against network intrusion detection systems. Digital Threats: Research and Practice, 2022, 3(3): Article 31. doi:10.1145/3469659.

[14] Li M, Xu Y, Li N, Jin Z. Adversarial attacks on deep learning-based methods for network traffic classification. In: Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2022: 1123–1128. doi:10.1109/TrustCom56396.2022.00154.

[15] Areia J, Bispo IA, Santos L, Costa RL de C. IoMT-TrafficData: Dataset and tools for benchmarking intrusion detection in Internet of Medical Things. IEEE Access, 2024, 12: 1–15. doi:10.1109/ACCESS.2024.3437214.

[16] Xiong WD, Luo KL, Li R. AIDTF: Adversarial training framework for network intrusion detection. Computers & Security, 2023, 128: 103141. doi:10.1016/j.cose.2023.103141.

[17] Alshahrani E, Alghazzawi D, Alotaibi R, Rabie O. Adversarial attacks against supervised machine learning-based network intrusion detection systems. PLOS One, 2022, 17(10): e0275971. doi:10.1371/journal.pone.0275971.

[18] Ci Z, Liu Z, Song Y, Qin F, Li Y, Zhao J. Multi-task class-aware adversarial training for remote sensing object detection robustness. Connection Science, 2025, 37(1). doi:10.1080/09540091.2025.2581373.

[19] Yan J, Liu Z, Xie Y, Liang S, Liu L, Xu K. CertTA: Certified robustness made practical for learning-based traffic analysis. In: Proceedings of the 34th USENIX Security Symposium, 2025: 7349–7368.

[20] Alhajjar E, Maxwell P, Bastian N. Adversarial machine learning in network intrusion detection systems. Expert Systems with Applications, 2021, 186: 115782. doi:10.1016/j.eswa.2021.115782.

[21] Vitorino J, Dias T, Fonseca T et al. Constrained adversarial learning for automated software testing: A literature review. Discover Applied Sciences, 2025, 7: 547. doi:10.1007/s42452-025-07073-3.

[22] Susilo B, Muis A, Sari RF. Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm. Sensors, 2025, 25(2): 580. doi:10.3390/s25020580.

[23] Quirumbay Yagual D, Fernández Iglesias D, Nóvoa FJ. A hybrid deep learning-based architecture for network traffic anomaly detection via EFMS-enhanced KMeans clustering and CNN-GRU models. Applied Sciences, 2025, 15(20): 10889. doi:10.3390/app152010889.

[24] Abiha UE, Rehman A, Abbas A, Haider MA, Al-Yarimi FAM, Gul MU, Hassan SR. Improving adversarial resilience for anomaly detection in the heterogeneous Internet of Things through ensemble models. Future Generation Computer Systems, 2026, 178: 108299. doi:10.1016/j.future.2025.108299.

[25] Dahanayaka T, Ginige Y, Huang Y, Jourjon G, Seneviratne S. Robust open-set classification for encrypted traffic fingerprinting. Computer Networks, 2023, 236: 109991. doi:10.1016/j.comnet.2023.109991.

[26] Xia L, Wang G. Research on IoT data anonymization method based on multimodal adversarial generative networks and reinforcement learning collaborative optimization. Journal of Cyber Security and Mobility, 2025, 14(04): 799–822.

[27] Heydari V, Nyarko K. Enhancing adversarial robustness in network intrusion detection: A novel adversarially trained neural network approach. Electronics, 2025, 14(16): 3249. doi:10.3390/electronics14163249.

[28] Bi S, Wang J, Song J, Li P, Li L. Research on the intrusion detection model for power Internet of Things combining deep belief network and BiLSTM. Journal of Cyber Security and Mobility, 2025, 14(03): 653–672.

[29] Baldoni S, Battisti F. Histogram-based network traffic representation for anomaly detection through PCA. Computer Networks, 2025, 265: 111276. doi:10.1016/j.comnet.2025.111276.

[30] University of Padua. Network Anomaly Detection. Available: https://medialab.dei.unipd.it/research-areas/forensics-and-security/network-anomaly-detection/.

[31] Dogra A. CIC-IoT Dataset 2023. Available: https://www.kaggle.com/datasets/akashdogra/cic-iot-2023.

[32] University of New South Wales. The Bot-IoT Dataset. Available: https://research.unsw.edu.au/projects/bot-iot-dataset.

[33] University of New South Wales. The TON_IoT Datasets. Available: https://research.unsw.edu.au/projects/toniot-datasets.

[34] Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 2015: 1–6. doi:10.1109/MilCIS.2015.7348942.

[35] Dongxu Sun, Peng Li, Lina Pei, Chuhang Wei. Online Network Intrusion Detection Based on Self-Encoder Integration. In: Proceedings of the 3rd International Conference on Signal Processing, Computer Networks and Communications (SPCNC '24). Association for Computing Machinery, New York, NY, USA, 2025: 246–250.

[36] Toba AL, Kulkarni S, Khallouli W, Pennington T. Long-term traffic prediction using deep learning long short-term memory. Smart Cities, 2025, 8(4): 126. doi:10.3390/smartcities8040126.

[37] Ma H, Zhang W, Zhang D et al. An IoT intrusion detection framework based on feature selection and large language models fine-tuning. Sci Rep, 2025, 15: 21158. doi:10.1038/s41598-025-08905-3.

[38] Gautam S, Malhotra A, Dhurandher SK. A hybrid CNN-LSTM model for enhanced intrusion detection in Internet of Things environments. In: 2025 International Conference on Computational, Communication and Information Technology (ICCCIT), Indore, India, 2025, 781–786, doi:10.1109/ICCCIT62592.2025.10927909.

[39] Xu S, Han J, Liu Y et al. Few-shot traffic classification based on autoencoder and deep graph convolutional networks. Sci Rep, 2025, 15: 8995. doi:10.1038/s41598-025-94240-6.

[40] Madry Aleksander et al. Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018 – Conference Track Proceedings, 2018. Available: https://openreview.net/forum?id=rJzIBfZAb.

[41] Tchaleu BY, Ndjiongue AR, Leke CA. Generative adversarial networks: A comprehensive review and the way forward. SAIEE Africa Research Journal, 2025, 116(3): 101–124. doi:10.23919/SAIEE.2025.11090063.

[42] Haiqi Zhang, Guangquan Lu, Mengmeng Zhan, Beixian Zhang. Semi-supervised classification of graph convolutional networks with Laplacian rank constraints. Neural Process Lett, 2022, 54(4): 2645–2656. doi:10.1007/s11063-020-10404-7.

[43] Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC.

Biographies

Chenxuan Li, master, lecturer, majors in Signal and Information Processing. His research focuses on information security, artificial intelligence, intelligent security detection and data privacy protection. Equipped with solid theoretical knowledge and professional research capabilities, he has taken part in multiple research projects and gained abundant practical experience. He devotes himself to advancing the integration of AI and information security.

Xinyi Zhang obtained her master's degree from Beijing Jiaotong University in 2017. Since 2017, She has been a full-time teacher at Zhengzhou Railway Vocational and Technical College. She is the author of three books, more than 10 articles, and more than eight inventions. Her research focuses on intelligent traffic network and transportation safety.