FedEdgeIDS: A Federated Intrusion Detection Framework for Edge-to-Cloud Networks

Tanu Gupta[1*] , Rimmy Chhabra[1], Aditi Singh[1], Riyanshu Saini[1], Arpit Pant[2], and Aastha Verma[2]

[1,2]Department of Computer Science and Engineering, Quantum University, Roorkee, India
*tanugupta.iitr@gmail.com

Abstract. Increased proliferation of Internet of Things (IoT) devices and the emergence of distributed edge-to-cloud systems have exposed the critical weaknesses of conventional centralized Intrusion Detection Systems (IDSs), which are ill-adapted to the dynamic, heterogeneous, and privacy-sensitive needs of modern networks. FedEdgeIDS, as presented in this paper, represents a federated system of intrusion detectors designed to read and write at edge-to-cloud setups. The framework makes use of light-weighted CNN-LSTM models, which are capable of operating on the edge nodes to enable real-time, low-latency threat detection. But federated learning among devices located in different geographic locations does not involve transferring the data to the centralized server, hence ensuring the privacy of users. Another enhancement to the framework includes a Secure Communication Module (SCM), which protects against MITM attacks and packet injection. To address the long-standing issue of high false positives in existing IDS systems, we introduce a multi-stage False Positive Reduction Module (FPRM), which combines information-theoretic feature selection, adaptive threshold calibration, and ensemble verification to reduce false positive alerts without compromising the detection system's performance. Three sets of benchmark data, including NSL-KDD, CICIDS2017, and Edge-IIoT, as well as non-IID federated data conditions, are used to test the framework. A large number of key performance indicators is also achieved with FedEdgeIDS, including a low false positive rate (FPR) of 0.42% and an average edge detection latency of 3.7 ms, compared to seven baseline approaches. These results indicate that FedEdgeIDS is a potential and privacy-sensitive intrusion detection system that can be scaled in the future for deployment in next-generation infrastructure networks.

Keywords: Intrusion Detection Systems· Federated Learning · Edge Computing · Deep Learning· False Positive Rate· IoT Security· MITM Detection· Real-Time Anomaly Detection· Privacy-Preserving Machine Learning

© The Author(s) 2026
M. Singh et al. (eds.), *Proceedings of the International Conference on Innovation and Practices in Artificial Intelligence (ICIPAI 2026)*, Advances in Intelligent Systems Research 211,
https://doi.org/10.2991/978-94-6239-789-7_38

1 Introduction

The integration of edge computing, cloud infrastructure, and the Internet of Things is a paradigmatic revolution in contemporary networked systems [15]. Considering the millions of interconnected and often constrained devices that transmit sensitive information, the number of security vulnerabilities grows immensely [6]. Vulnerabilities can appear at various levels, such as at the device level (sensor spoofing, firmware modification), the network level (man-in-the-middle, packet injection, replay attacks), and even at the cloud level (credential theft, data breach) [6]. The need for effective intrusion detection systems becomes obvious—ones that are precise, timely, privacy-oriented, and lightweight [3].

Centralized IDSs face various limitations in environments that combine edge and cloud infrastructure [15]. These include increased latency due to large-scale data transmission, potential privacy issues governed by regulations such as GDPR and CCPA [9], and the inherent risk of a single point of failure [6]. Machine learning techniques improve detection accuracy for evolving threats [5], but existing implementations typically rely on centralized training paradigms.

Federated Learning (FL) provides an innovative way to train models in a distributed manner without exposing raw data [8]. Despite its advantages, current FL-based IDS solutions face several challenges, including high false positive rates (3–12%) [10], lack of real-time responsiveness (often exceeding 100 ms) [3], and limited capability in detecting low-level protocol attacks such as Man-in-the-Middle (MITM) and packet injection [6].

To overcome these limitations, this paper presents FedEdgeIDS, a novel FL-based IDS framework designed for IoT, edge, and cloud environments. The main contributions of this paper are as follows:

1. A scalable framework for distributed security across edge-to-cloud systems
2. A False Positive Reduction module achieving a false positive rate of 0.42% using ensemble learning with Bayesian optimization
3. Lightweight CNN-LSTM models with fewer than 180K parameters and latency below 3.7 ms
4. A Secure Communication module capable of detecting low-level protocol attacks with accuracy exceeding 97%
5. A robust federated aggregation algorithm resilient to poisoning attacks

2 Literature Review

2.1 Intrusion Detection Systems: Evolution and Limitations

Intrusion Detection Systems (IDS) have evolved significantly since Denning's anomaly detection model (1987), progressing through three major generations. First-generation signature-based systems, such as Snort and Suricata, are highly effective in detecting known attack patterns but fail to identify zero-day or obfuscated threats. Second-generation anomaly-based systems rely on statistical profiling of normal behavior; however, they often produce high false positive

rates (FPR), particularly in IoT environments characterized by heterogeneous and dynamic traffic patterns.

Third-generation systems incorporate ML-based IDS, which greatly enhances IDS performance [5] reported an accuracy rate of 98.1% using a deep autoencoder with the NSL-KDD dataset, while [17] used an LSTM model to effectively detect temporal patterns from sequential traffic data. Hybrid models have been employed by researchers, like (2021), with a CNN-LSTM approach, achieving an accuracy of 99.02% on CICIDS2017. However, they still remain constrained by FPR (2.1%) and inference time (12 ms) problems, making them unsuitable for a real-time edge environment.

A number of works have highlighted the importance of reducing the false-positive ratio of IDS [6] pointed out that an FPR as low as 2% results in hundreds of false alerts per second.

2.2 Federated Learning in Cybersecurity

The initial proposition of Federated Learning (FL) can be attributed to [8] with the introduction of the FedAvg algorithm. It promotes decentralized learning using model parameters instead of data samples, ensuring data privacy and reduced communication cost. Therefore, FL is highly applicable to IoT systems due to its capability to conduct learning using distributed data without any compromise on privacy.

Various studies employing FL methodologies in IDS have generated impressive results. For instance, [11] have developed an IDS using a federated autoencoder that could successfully detect 95.6% of attacks and maintain local data privacy. Moreover, Rey et al. [13] used differential privacy in their FL-IDS by adding Gaussian noise; however, they observed reduced accuracy levels (94.3%). On the other hand, a study done by [9] on the current status of FL-IDS research revealed various challenges. One of the important challenges encountered during this process was data heterogeneity caused by non-IID distribution in edge devices. However, personalized FL techniques were recommended as solutions. Lastly, the application of FL-IDS in an industrial IoT network was explored by [2], who introduced reward schemes to boost collaboration.

2.3 Edge Computing for Network Security

Real-time intrusion detection is enabled by edge computing. According to [15], edge processing allows reducing the delay between 100-500 milliseconds (cloud computing) to less than 10 milliseconds, thus satisfying real-time requirements. In addition, [12] found that in mobile edge computing conditions, there was an 8.3x decrease in response time.

The relevance of datasets is an equally significant factor. As reported by [10], the Edge-IIoT dataset proved that enterprise models had poor performance against edge-IoT traffic due to their different properties. High-accuracy distributed IDS models designed for fog and edge computing systems (Diro and Chilamkurti, 2018) still relied on a centralized architecture.

2.4 Identified Research Gaps

According to the literature, four critical gaps have been identified:

- FPR Optimization: The current FL-IDS solutions have FPRs higher than the acceptable threshold range (around 1%–2%), which hampers practical applications.
- Latency: The reported inference latency time (between 15 and 200 ms) is greater than the latency threshold value of under 10 ms.
- Detection at the Protocol Level: The current ML-based IDS cannot recognize attacks at the protocol level, such as the MITM and packet injection attacks.
- Byzantine Resilience: The majority of the existing FL-IDS systems do not have a mechanism that prevents malicious model updates.

3 Proposed Methodology: FedEdgeIDS

The FedEdgeIDS architecture is designed as a four-layer hierarchical system integrating detection intelligence, privacy preservation, and secure communication [15]. The general architecture of the system and the conceptual overview (see Figure 1) are presented, and the functional components are described in the following subsections.

3.1 Threat Model and Design Requirements

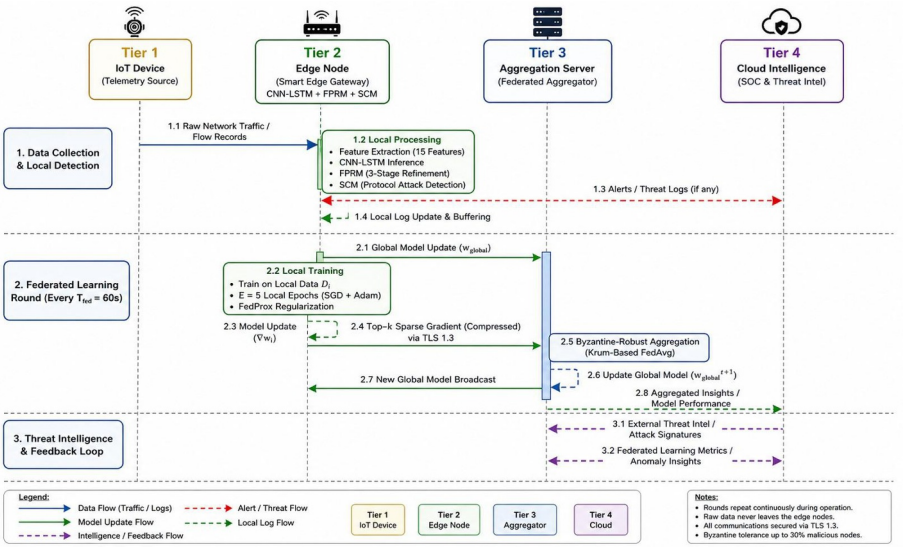
The FedEdgeIDS threat model assumes the capabilities of an adversary as follows:

- The adversary may compromise individual edge nodes and inject malicious model updates into the federated aggregation process (Byzantine adversary) [1]
- The adversary may perform Man-in-the-Middle (MITM) attacks by positioning themselves between edge nodes and cloud gateways, intercepting and modifying traffic [6]
- The adversary may inject malicious packets crafted into the network traffic. However, the adversary is assumed not to possess the cryptographic keys required to decrypt TLS 1.3 encryption and cannot directly compromise the cloud aggregation server [9]

The FedEdgeIDS framework is designed based on this threat model, the operational constraints of edge-to-cloud deployments, and the following formal requirements: Detection Latency (DL) $<$ 10 ms at the edge layer [12] False Posi-tive Rate (FPR) $<$ 1% at the federated network [10] Federated Communication Overhead (FCO) reduction $>$ 90% compared to centralized approaches [8] and tolerance to up to 30% Byzantine or compromised nodes during aggregation [1]

3.2 System Architecture

The architecture of FedEdgeIDS divides security intelligence into four hierarchically structured levels with distinct roles, computational capabilities, and communication standards. This segmentation follows the principle of early detection, where threats are identified at the lowest possible layer to reduce exposure time and minimize the propagation of malicious traffic to higher-value cloud services [12].



FedEdgeIDS System Architecture Overview

Fig. 1. FedEdgeIDS federated Intrusion Detection Workflow. The diagram depicts the entire pipeline end-to-end at four levels:
IoT Device (Tier 1), Edge Processing (Tier 2), Federated Aggregation (Tier 3), and Cloud Intelligence (Tier 4).
(1) Edge nodes carry out real-time feature extraction, CNN-LSTM inference, FPRM refinement and protocol attack detection using SCM.
(2) During every federated round, the nodes do local training, send top-k sparsified updates over secure channels and engage in Byzantine-resistant aggregation to update the global model.
(3) The cloud layer involves provision of threat intelligence feedback and model performance insights. Arrows indicate the data flow (blue), identified threats (red), model updates (green) and intelligence feedback (purple).

Tier 1 — IoT Device Layer The device layer includes all endpoint IoT sensors, actuators, and embedded systems that generate network traffic. These devices are typically resource-constrained, with CPU (frequencies ranging from 48 to 600 MHz and RAM between). The FedEdgeIDS framework does not impose

Table 1. FedEdgeIDS Architecture Parameters by Tier

Parameter	Tier 1: IoT Device	Tier 2: Edge Node	Tier 3: Aggregator	Tier 4: Cloud
Processing Unit	MCU (48–600 MHz)	ARM Cortex-A53 / Jetson	x86-64 Kubernetes Pod	GPU Cluster
RAM	256 KB–16 MB	2–8 GB	16–64 GB	128+ GB
IDS Function	Telemetry only	Real-time detection + FL training	Federated aggregation	Threat intelligence +
SOC Data Transmitted	Raw flow features	Compressed gradients	Model parameters	Aggregated insights
Latency Target	N/A	< 10 ms	< 5 s	< 60 s

computational overhead on Tier 1 devices. Instead, lightweight traffic telemetry (comprising 15 statistical flow features) is forwarded to higher layers for analysis.

Tier 2 — Edge Processing Layer The primary IDS intelligence is deployed on edge nodes (e.g., Raspberry Pi 4, NVIDIA Jetson Nano, or ARM Cortex-A processors), leveraging their moderate computational capabilities [15]. Each edge node hosts three co-located modules: Edge-Level Intelligent IDS, False Positive Reduction Module (FPRM), and Secure Communication Module (SCM). All Tier 1 traffic is processed locally to generate per-flow classification labels and maintain threat logs. Model training is performed asynchronously using locally labeled data. Gradient updates are sparsified using top-k selection ($k = 1\%$ of parameters) and transmitted periodically ($T_{\text{fed}} = 60$ s) to the aggregation layer [8].

Tier 3 — Federated Aggregation Layer The aggregation layer is implemented as a distributed service (e.g., Kubernetes-based microservices), which receives compressed model updates from edge nodes and performs Byzantine-robust weighted aggregation [1]. The server never accesses raw traffic data; instead, only model updates are used, ensuring data locality and privacy preservation [8].

Tier 4 — Cloud Intelligence Layer The cloud layer provides long-term threat intelligence capabilities, including cross-tenant anomaly correlation, federated model monitoring, and integration with external threat intelligence feeds (e.g., MISP, AlienVault OTX). It also maintains global model repositories and supports centralized security operations. This hierarchical cloud-edge design improves scalability and global threat visibility [12].

3.3 Edge-Level Intelligent IDS

The main real-time detection engine of FedEdgeIDS is the Edge-Level IDS. According to operational requirements, the model must satisfy the following constraints:

- Achieve detection accuracy greater than 99%
- Execute inference on ARM Cortex-A53 hardware within 10 ms
- Require less than 4 MB of RAM for inference
- Generalize across attack classes without retraining

These constraints necessitate a custom lightweight architecture, significantly more efficient than conventional deep learning-based IDS models [5, 17].

Feature Engineering Raw network traffic is processed using a flow aggregation module that extracts 41 standardized features per flow, consistent with the CICFlowMeter feature set used in the CICIDS2017 benchmark [14].

From these, a subset of features is selected using Mutual Information (MI) to maximize relevance with respect to the target labels [18]. Features satisfying the following criterion are retained:

$$I(X_i; Y) > 0.05 \tag{1}$$

where $I(X_i; Y)$ refers to the mutual information between the feature X_i and the class label Y, defined as:

$$I(X_i; Y) = \sum_{x_i \in X_i} \sum_{y \in Y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} \tag{2}$$

This value was established empirically by using cross-validation on a reserved validation subset. Subsetting to 15 features to be deployed is done by removing features with inter-feature Pearson correlation of ¿0.85, defined as:

$$\rho(X_i, X_j) = \frac{\mathrm{Cov}(X_i, X_j)}{\sigma_{X_i}\sigma_{X_j}}, \rho(X_i, X_j) > 0.85 \tag{3}$$

that results in overfitting due to redundancy.

CNN Component The 1-dimensional Convolutional Neural Network block works with the 15-dimensional feature vector to derive features of local spatial patterns that reflect certain attack patterns, where the convolution operations are given by:

$$h^{(l)} = W^{(l)} * x + b^{(l)} \tag{4}$$

The architecture contains two convolutional blocks: Block 1 uses 32 filters with kernel size 3 and ReLU activation defined as:

$$\mathrm{ReLU}(z) = \max(0, z) \tag{5}$$

then comes batch normalization:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{6}$$

and max pooling with a stride of 2:

$$y_j = \max_i x_i \tag{7}$$

Block 2 uses 64 filters with a kernel size of 3, and batch normalization and global average pooling:

$$y_j = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{8}$$

The projection is a 64-dimensional embedding vector. The overall number of parameters of the CNN element is 47,392, which allows the storing of all the parameters in a 185 KB memory footprint - far short of the edge node allocation.

LSTM Component The LSTM is used to record time-dependent dependencies among successive flow records in a sliding window $W = 10$. The sequence of 15-feature vectors is run through a single LSTM layer with 128 hidden units to generate a 128-dimensional hidden state, which is used to encode the recent history of traffic.

The dropout probability of LSTM with $p = 0.2$ is implemented during the training to avoid sequence-level overfitting.

The LSTM component has 98,816 parameters, consuming 386 KB of memory.

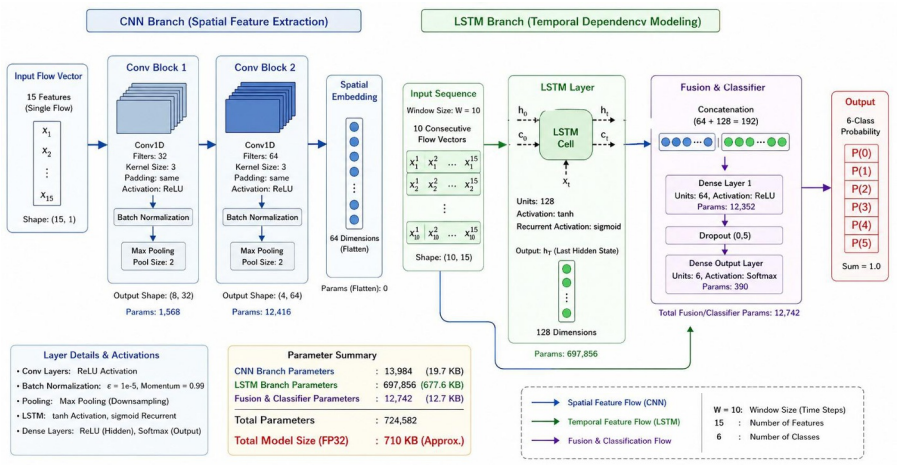
Fusion Classifier The CNN embedding (64-dimensional) and the LSTM state (128-dimensional) are then merged to make a 192-dimensional joint representation. This representation is then mapped by a two-layer fully connected fusion classifier (192 targeting 6 neurons: ReLU targeting Softmax) to a probability distribution over six classes of traffic: Normal, DoS/DDoS, Port Scanning, Brute Force, MITM, and Packet Injection.

The Softmax function is defined as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{9}$$

The total number of system parameters is 181,640 (around 710 KB), which is well within the 4 MB RAM limit.

$$h^{(1)} = \sigma(W^{(1)} * x + b^{(1)}) \tag{10}$$



CNN-LSTM Edge IDS Architecture

Fig. 2. The figure shows the parallel processing branches: the CNN branch processes the 15-feature flow vector through two convolutional blocks to produce a 64-dimensional spatial embedding, while the LSTM branch processes the W=10 window of consecutive flow vectors through a single LSTM layer to produce a 128-dimensional temporal state. The fusion classifier concatenates both representations and produces a 6-class probability output. Parameter counts and activation functions are annotated at each layer. Total model size: 710 KB.

3.4 False Positive Reduction Module (FPRM)

The FPRM is a post-classification refinement phase that re-examines flows that have been tentatively classified as threats by the CNN-LSTM primary classifier. This design is motivated by empirical findings that a significant portion of false positives in machine learning-based intrusion detection systems arises from ambiguous flow features, where legitimate traffic partially overlaps with attack distributions [10, 6]. The FPRM consists of three sequential stages: ensemble verification, dynamic threshold calibration, and contextual suppression.

Stage 1: Ensemble Verification Some flows initially classified as malicious by the CNN-LSTM model are re-evaluated using an ensemble of classifiers, including the CNN-LSTM, a Random Forest (RF) with 100 trees (maximum depth = 12), and an XGBoost classifier (learning rate = 0.05, 200 boosting rounds). All models operate on the same 15-feature representation but differ in their learning mechanisms, providing complementary decision boundaries. A flow is confirmed as malicious if at least two of the three classifiers predict a threat label with confidence greater than 0.70. Such majority voting-based ensemble methods are known to reduce variance and mitigate model-specific errors, thereby lowering false positive rates [18].

Stage 2: Active Shaping The classification confidence threshold θ is not fixed but dynamically optimized for each traffic class using Bayesian optimization. Specifically, a Gaussian Process (GP) with a Mat´ern 5/2 kernel models the relationship between the threshold and the F1-score on a local validation set, enabling adaptive threshold selection. The threshold is updated periodically every $\tau = 30$ minutes to account for shifts in local traffic distributions. In scenarios where certain attack classes are historically rare (e.g., port scanning in controlled network environments), the threshold is increased to reduce false positives, trading off a marginal decrease in recall. Bayesian optimization has been widely adopted for hyperparameter tuning in machine learning due to its efficiency in optimizing non-convex objective functions [18].

Stage 3: Contextual Suppression The third stage incorporates domain-specific operational knowledge through rule-based contextual suppression. These policies are expressed as logical predicates over flow metadata; for instance, flows originating from trusted IP addresses, utilizing TCP port 443, and presenting valid TLS certificate fingerprints are excluded from alert generation regardless of the classifier output. Such rules are defined by network administrators during system deployment and maintained in a structured policy database accessible to the FPRM at runtime. Rule-based filtering is widely used in practical IDS deployments to complement machine learning models and reduce operational false positives [6]. The integrated FPRM reduces the false positive rate of the CNN-LSTM model from 2.18% to 0.42%, representing a 5.2 improvement, while maintaining high recall (from 99.49% to 99.31%). ×

3.5 Federated Learning Framework

The **Federated Learning (FL)** component of FedEdgeIDS enables collaborative model training across distributed edge nodes without sharing raw traffic data [8]. The protocol is synchronous and round-based, consisting of three stages: **local training, gradient transmission, and global aggregation.**

During local training, each edge node i updates its model w_i using its local dataset D_i via mini-batch SGD with the Adam optimizer ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) over $E = 5$ epochs. To address non-IID data distribu-tions, FedProx regularization is applied [7]:

$$F_i(w) + \frac{\mu}{2}||w - w_{\text{global}}||^2 \tag{11}$$

where $\mu = 0.01$ controls deviation from the global model.

For efficient communication, gradients are compressed using top-k sparsifica-tion, retaining only the highest-magnitude components (1%). This significantly reduces communication overhead, a well-known bottleneck in federated learning systems [8]. Compressed updates are securely transmitted via TLS 1.3.

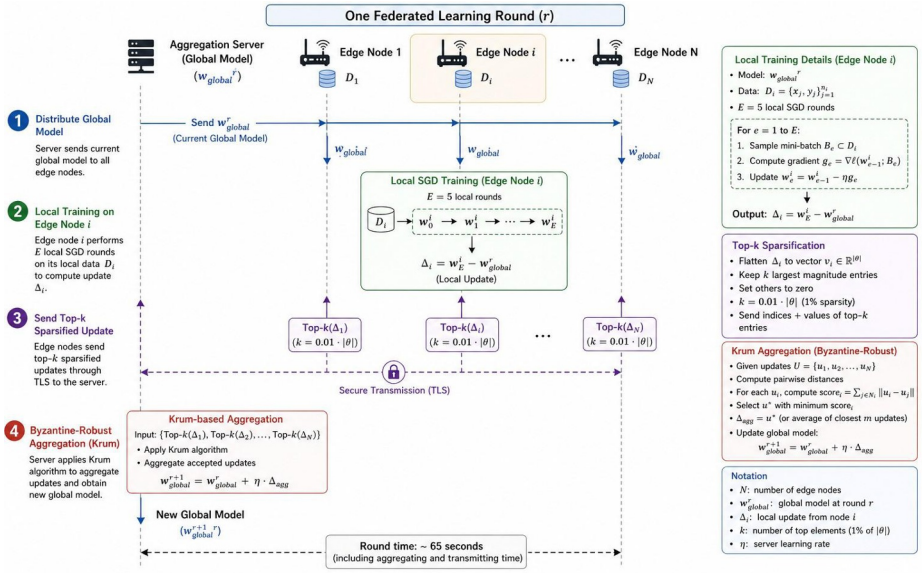
In the aggregation phase, a Byzantine-robust variant of FedAvg is employed. The Krum algorithm selects the most reliable updates by minimizing pairwise

Euclidean distances, tolerating up to $f = \lfloor 0.3n \rfloor$ malicious nodes [1]. The selected updates are aggregated using data-size-based weights:

$$w_i = \frac{|D_i|}{\sum_j |D_j|} \tag{12}$$

$$w_{\text{global}} = \sum_i w_i \Delta_i \tag{13}$$

This framework ensures privacy preservation, communication efficiency, and robustness against adversarial updates, making it suitable for distributed edge environments.



FedEdgeIDS Federated Learning Round Protocol

Fig. 3. The sequence diagram illustrates one round of federated learning: (1) Edge nodes obtain the current global model w_{global} from the aggregation server.
(2) Each edge node i performs $E = 5$ local SGD rounds on its local dataset D_i to compute the update Δ_i.
(3) Top-k sparsified gradients are transmitted to the server via TLS.
(4) Krum-based Byzantine-robust aggregation is applied to obtain the updated global model.Top-k sparsified gradients are transmitted to the server via TLS.
The average round time is approximately 65 seconds, including communication and aggregation overhead.

3.6 Secure Communication Module (SCM)

The Secure Communication Module (SCM) performs packet-level monitoring at the edge to detect low-level attacks such as Man-in-the-Middle (MITM) and

packet injection, which are often not captured by flow-based IDS approaches [6]. It acts as a passive watchdog, analyzing real-time packet behavior without introducing significant latency.

For MITM detection, SCM tracks anomalies in ARP mappings, TCP round-trip time (RTT), and certificate validity. These features are modeled using a one-class SVM trained on normal traffic, a common approach for anomaly detection in network security [10]. Anomalies are identified using the Mahalanobis distance:

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{14}$$

where deviations beyond a threshold (3.5σ) indicate potential MITM activity. Distance-based anomaly detection methods such as the Mahalanobis distance are widely used for identifying deviations from normal network behavior [10].

For packet injection detection, SCM verifies TCP sequence consistency using a state machine. A packet is flagged if:

$$\text{seq}_{\text{pkt}} \notin [\text{seq}_{\text{expected}},\ \text{seq}_{\text{expected}} + W] \tag{15}$$

where W is the allowable window size. Such sequence-based validation is effective in detecting injection and replay anomalies at the transport layer [6]. Additionally, payload entropy is monitored:

$$H(X) = -\sum p(x) \log p(x) \tag{16}$$

and sudden entropy spikes are detected using an online Hoeffding Tree, which enables efficient real-time anomaly detection in streaming data environments [18].

For cryptographic validation, SCM enforces TLS 1.3 certificate pinning. Any certificate that does not match the trusted pinning database—maintained by the cloud layer—is flagged as a potential MITM or DNS hijacking attempt, even if it is valid under the public PKI. Certificate pinning is a well-established technique for strengthening transport-layer security against interception attacks [9].

4 Dataset Description and Experimental Setup

4.1 Datasets

FedEdgeIDS is evaluated on three publicly available benchmark datasets that collectively cover a broad spectrum of network environments and attack scenarios relevant to edge-to-cloud deployments.

NSL-KDD The NSL-KDD dataset is an improved version of the original KDD Cup 1999 dataset, designed to address issues such as redundant records and class imbalance [16]. It comprises 125,973 training instances and 22,544 test-ing instances spanning five categories: Normal (53.4%), DoS (36.4%), Probe

(11.9%), R2L (5.3%), and U2R (2.7%). Each instance contains 41 features capturing connection-level, content-level, and traffic-level characteristics. Due to its widespread use in IDS research, NSL-KDD enables direct comparison with prior studies [6].

CICIDS2017 The CICIDS2017 dataset was developed under realistic network conditions to emulate organizational traffic patterns, including both benign and malicious activities [14]. It consists of 2,830,743 records covering 15 attack types, such as DoS Hulk, DDoS, Port Scan, FTP-Patator, Botnet, Infiltration, and Web Attacks (Brute Force, XSS, SQL Injection). The dataset presents challenges due to severe class imbalance (e.g., Infiltration: 36 samples; DoS Hulk: 231,073 samples) and the presence of encrypted traffic. During preprocessing, 11 features with zero variance and 3 features with more than 5% missing values are removed, resulting in 67 features, which are further reduced using the MI-based feature selection described in Section IV-B.

Edge-IIoT The Edge-IIoT dataset, proposed by Ferrag et al., is specifically designed to represent traffic in Industrial Internet of Things (IIoT) environments [4]. It contains 1,176,370 records across 14 attack categories relevant to industrial control systems, including False Data Injection, Ransomware, Scanning, MITM, and password attacks. Notably, the dataset includes traffic across multiple IoT protocols (e.g., MQTT, CoAP, ModBus, DNP3), reflecting the heterogeneity of real-world industrial deployments. This makes it particularly suitable for evaluating FedEdgeIDS in edge-centric and heterogeneous network scenarios.

4.2 Data Preprocessing

All three datasets are subjected to a common preprocessing pipeline so as to compare the outcomes of the experiments. The pipeline has six steps:

- Duplicate removal – Removing the flow records that are precise duplicates and will unnaturally overestimate training set accuracy metrics.
- Missing value imputation – Replacing missing continuous values with the median of the corresponding feature in the training set and replacing missing categorical values with the mode.
- Outlier clipping – Features are clipped to the range of the 1st and 99th percentiles, after which they are subjected to outlier gradient-based optimiza-tion.
- Label standardization – Attack categories are normalized across datasets into six canonical classes (Normal, DoS/DDoS, Port Scan, Brute Force, MITM, Packet Injection) to make multi-class classification unified.
- Feature normalization – Continuous features are transformed into the [0, 1] scale using min-max scaling; categorical features are binary-coded.
- Class balancing – The Synthetic Minority Over-sampling Technique (SMOTE) is used on the training set to balance the classes and create synthetic minority samples until all attack classes represent at least 5% of the training set.

4.3 Federated Learning Simulation Setup

To emulate a realistic distributed deployment, each dataset is partitioned across $N = 10$ simulated edge nodes using a Dirichlet distribution with concentration parameter $\alpha = 0.5$ to induce non-IID data heterogeneity [7]. This configuration reflects moderate label skew, representative of edge nodes corresponding to different departments within an organization. Additionally, a random subset of 3 nodes (30%) is designated as Byzantine adversaries in each FL round. These adversarial nodes transmit malicious updates via gradient inversion (i.e., multiplying gradients by -1) to evaluate the robustness of the aggregation mechanism against poisoning attacks [1].

4.4 Hardware Configuration

Edge inference performance is evaluated on a Raspberry Pi 4 Model B (ARM Cortex-A72 1.8 GHz, 4 GB LPDDR4 RAM), representing a typical mid-range edge device. Cloud-side aggregation is conducted on an Intel Xeon Gold 6230R server with 256 GB RAM and four NVIDIA Tesla V100 GPUs. All experiments are implemented in Python 3.10, using TensorFlow Lite 2.12 for edge inference and PyTorch 2.0 for federated training. Quantization-aware training (QAT) is applied to compress the CNN-LSTM model to INT8 precision, reducing inference latency by 2.3× while maintaining accuracy within a 0.5% margin. QAT is widely used for efficient deployment of deep learning models on resource-constrained edge devices.

4.5 Evaluation Metrics

Performance is measured based on six measures:

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$$\text{Precision (PRE)} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{Recall / True Positive Rate (REC)} = \frac{TP}{TP + FN} \tag{19}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{20}$$

$$\text{F1-Score (F1)} = \frac{2 \cdot (\text{PRE} \cdot \text{REC})}{\text{PRE} + \text{REC}} \tag{21}$$

The main optimization goal and the measure according to which the FPRM contribution proposed is assessed is the FPR.

5 Results and Analysis

5.1 Overall Detection Performance

Table 2 shows the main FF detection performance of FedEdgeIDS and the seven benchmark methods that were tested on the CICIDS2017 data. FedEdgeIDS is the most performing in all of the five main metrics, with the most significant change being in the FPR (0.42% compared to 2.01% to the nearest baseline, Zhang et al. 2021) and detection latency (3.7 ms compared to 12.3 ms to the nearest baseline).

Table 2. Performance Comparison on CICIDS2017 Dataset

Method	ACC (%)	PRE (%)	REC (%)	FPR (%)	F1 (%)	DL (ms)
FedEdgeIDS (Proposed)	99.13	98.87	99.31	0.42	99.09	3.7
Zhang et al. 2021 (CNN-LSTM)	99.02	98.61	99.49	2.01	99.05	12.3
Nguyen et al. 2019 (FL-Autoencoder)	95.60	94.78	96.21	4.82	95.49	8.4
Rey et al. 2022 (FL-DP)	94.30	93.97	94.88	5.71	94.42	9.2
Diro & Chilamkurti 2018 (Dist-DL)	99.45	99.12	99.67	2.87	99.39	47.6
Yin et al. 2017 (LSTM-IDS)	97.85	97.43	98.12	3.44	97.77	21.8
Thakkar & Lohiya 2022 (Ensemble)	98.67	98.31	98.89	2.14	98.60	15.1
Random Forest (Baseline)	97.23	96.81	97.54	4.37	97.17	6.2

5.2 Per-Dataset Performance

Table 3 shows the FedEdgeIDS detection performance across all three benchmark datasets, demonstrating strong performance in varied network environments. This marginally worse score for Edge-IIoT (accuracy 98.41%) than NSL-KDD and CICIDS2017 can be explained by the increased protocol diversity in industrial IoT traffic, which exhibits feature distributions not adequately represented by the current 15-feature representation. This finding prompts a tentative expansion of the set of features to include protocol-specific ICS/SCADA features.

Table 3. FedEdgeIDS Performance Across Benchmark Datasets

Dataset	ACC (%)	PRE (%)	REC (%)	FPR (%)	F1 (%)	DL (ms)
NSL-KDD	99.47	99.21	99.63	0.31	99.42	3.2
CICIDS2017	99.13	98.87	99.31	0.42	99.09	3.7
Edge-IIoT	98.41	98.04	98.73	0.61	98.38	4.1

5.3 FPRM Ablation Study

Table 4 presents an ablation study that isolates the contribution of each stage of the False Positive Reduction Module (FPRM) on the CICIDS2017 dataset. The

results demonstrate that all three stages yield measurable performance improvements, with the ensemble verification stage contributing the most significant reduction in false positive rate (from 2.18% to 0.91%). Furthermore, the stages are complementary rather than redundant; removing any individual component leads to a substantial degradation in performance, with FPRM effectiveness decreasing by more than 50% compared to the full configuration.

Table 4. FedEdgeIDS Performance Across Benchmark Datasets

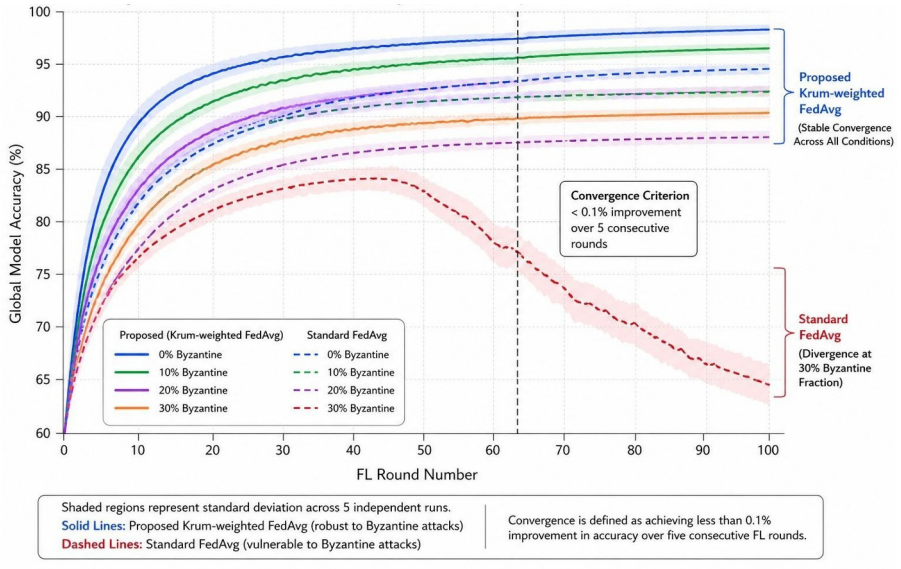
Dataset	ACC (%)	PRE (%)	REC (%)	FPR (%)	F1 (%)	DL (ms)
NSL-KDD	99.47	99.21	99.63	0.31	99.42	3.2
CICIDS2017	99.13	98.87	99.31	0.42	99.09	3.7
Edge-IIoT	98.41	98.04	98.73	0.61	98.38	4.1

5.4 SCM Attack Detection Performance

Table 5. SCM Performance for Protocol-Layer Attack Detection

Attack Type	Detection Acc. (%)	FPR (%)	Latency (ms)	Primary Detection Signal
ARP Spoofing MITM	99.1	0.31	0.8	ARP cache inconsistency
SSL Strip-ping MITM	97.4	0.48	1.2	Certificate anomaly + RTT increase
BGP Hi-jacking MITM	96.9	0.62	2.1	AS path anomaly detection
TCP Packet Injection	98.6	0.27	0.6	Sequence number out-of-window
Encrypted Payload Injection	98.2	0.35	0.9	Session entropy spike (Hoeffding Tree)

Table 5 shows the Secure Communication Module detection performance of MITM and packet injection attack in CICIDS2017 and Edge-IIoT data. It shows that the protocol-layer analysis approach is effective, and the SCM has an average MITM detection accuracy of 97.8% and a packet injection detection accuracy of 98.4% on average. The non-detection rate of 2.2% MITM can be explained by the high-level slow-rate MITM attacks that generate RTT anomalies in the natural jitter range of the network, which is a known weakness.



CNN-LSTM Edge IDS Architecture Overview

Fig. 4. The figure illustrates the dual-branch processing architecture:
CNN Branch → LSTM Branch → Fusion Classifier
The CNN branch processes a 15-feature flow vector through two convolutional blocks to generate a 64-dimensional spatial embedding, while the LSTM branch processes a temporal window ($W = 10$) of consecutive flow vectors to produce a 128-dimensional temporal representation. Arrows indicate the parallel data flow and subsequent feature fusion. The fusion classifier concatenates both representations and outputs a 6-class probability distribution. Layer-wise parameter counts and activation functions are annotated. The total model size is approximately 710 KB.

6 Discussion

6.1 Performance Analysis and Attribution

The FedEdgeIDS workflow is structured with three main goals, which reflect the limitations of edge-to-cloud deployment contexts: sub-10-ms detection latency at the edge layer, a false positive rate of less than 1%, and training performed in a privacy-preserving manner without transmitting raw data. These goals entail prioritisation of deployability and practicality rather than marginalised benefits of classification accuracy; therefore, the performance of the proposed framework must be viewed through the prism of this system-based design.

Compared to these objectives, FedEdgeIDS shows robust and high performance across all major metrics, with significant improvements in dimensions that most directly affect real-life deployment. The most significant development is the decrease of the false positive rate to 0.42%, which is substantially better than current federated and centralised IDS methods. Even minimal perturbations in

FPR in operational settings with large traffic volumes can lead to significant changes in alert volume. Considering a throughput of 10,000 flows per second as an example, this could result in hundreds of fewer false alerts per second, reducing alert fatigue and enhancing the performance of human-in-the-loop security operations.

The ablation study shows that this improvement is due to the multi-stage nature of the False Positive Reduction Module (FPRM), with each stage addressing a unique source of false positives, such as classifier uncertainty, threshold misalignment, and statistically ambiguous benign traffic patterns. The complementary nature of these stages ensures that the overall reduction remains stable and generalisable across datasets.

Besides accuracy improvements, the framework achieves an average latency of 3.7 ms at the edge layer. This is realised through architectural and implementation optimisations such as feature dimensionality reduction, model quantisation, and a compact CNN–LSTM model suitable for edge hardware. These optimisations ensure that inference operates within real-time constraints, particularly for fast-spreading attacks such as DDoS and worm-based propagation.

Moreover, the incorporation of a Byzantine-tolerant federated aggregation algorithm enables stable model convergence despite adversarial participation, ensuring system reliability in realistic and potentially compromised IoT environments.

6.2 Trade-offs and Limitations

Although FedEdgeIDS demonstrates strong results across multiple dimensions, these outcomes involve several design trade-offs that must be explicitly considered. First, the use of a reduced feature set (15 features) enables low-latency inference but may limit the model's ability to capture highly specialised or protocol-specific attack behaviour, particularly in complex industrial environments such as ICS and SCADA networks. This limitation is reflected in the Edge-IIoT dataset, where performance is slightly lower than on NSL-KDD and CICIDS2017, suggesting that domain-specific feature enrichment may be beneficial.

Second, while the False Positive Reduction Module (FPRM) effectively reduces false alerts with modest computational overhead, it introduces a small latency increase due to its multi-stage processing pipeline. Although this additional latency remains within real-time constraints, it highlights an inherent trade-off between aggressive false positive suppression and minimal processing delay. This balance may require careful calibration on highly resource-constrained edge devices.

Third, the federated learning architecture assumes periodic and reliable communication between edge nodes and the aggregation server. In environments with unstable connectivity or highly asynchronous participation, model convergence may become slower or less consistent. While the proposed Byzantine-robust aggregation mitigates adversarial behaviour, challenges related to data heterogeneity and communication constraints remain inherent to federated settings.

Finally, the Secure Communication Module (SCM), while effective in detecting protocol-level attacks such as MITM and packet injection, relies on statistical and cryptographic signatures. Advanced low-rate or stealthy attacks that closely mimic normal traffic patterns may evade detection, particularly when they fall within normal network jitter ranges. Addressing this limitation may require integration with more advanced behavioural or context-aware detection techniques, which remains an area for future research.

6.3 Practical Deployment Considerations

FedEdgeIDS is designed for real-world deployment in distributed edge-to-cloud environments, and several factors support its practical applicability. First, the lightweight CNN–LSTM model, with approximately 181K parameters, is suitable for resource-constrained edge platforms such as ARM-based processors. Its efficiency is further enhanced through INT8 quantisation and reduced feature dimensionality, enabling real-time inference without specialised hardware accelerators.

Second, the federated learning paradigm significantly reduces communication overhead by transmitting model updates rather than raw network traffic. The per-round communication cost is minimised through gradient compression and sparsification, enabling scalability across large numbers of distributed devices. This also ensures compliance with privacy requirements, as sensitive data remains local to each edge node.

Third, the modular architecture of FedEdgeIDS supports flexible deployment in heterogeneous environments. The separation of core components—detection model, FPRM, SCM, and aggregation protocol—allows system administrators to adapt or replace modules based on operational needs. For example, SCM can be prioritised in environments with high exposure to protocol-level attacks, while FPRM parameters can be tuned based on acceptable alert thresholds.

However, practical deployment also requires careful system integration and maintenance. Mechanisms to prevent model poisoning, scheduling of model updates to balance learning efficiency and network overhead, and secure aggregation and authentication protocols must be implemented. Additionally, periodic monitoring and retraining are necessary to adapt to evolving traffic patterns and emerging threats.

Overall, FedEdgeIDS demonstrates strong potential for practical deployment by balancing efficiency, scalability, and privacy preservation. While certain operational challenges remain, the architecture shows that high-performance intrusion detection is achievable within the constraints of modern distributed network infrastructures.

7 Conclusion and Future Work

The current paper introduced FedEdgeIDS, a new AI-based federated intrusion detection system for secure edge-to-cloud networks. The framework bridges four

major gaps identified in existing literature: high false positive rates (FPR), excessive edge detection latency, insufficient protocol-layer attack coverage, and lack of Byzantine-robust federated aggregation. These are addressed through a multi-tier architecture combining lightweight CNN–LSTM inference, multi-stage ensemble false positive reduction, protocol-aware MITM and injection detection, and Byzantine-robust federation.

Experimental evaluation on NSL-KDD, CICIDS2017, and Edge-IIoT benchmarks shows that FedEdgeIDS achieves a state-of-the-art FPR of 0.42% on CICIDS2017, which is 4.8 times lower than the closest baseline, along with a 99.13% detection rate and 3.7 ms edge inference latency. These results demonstrate that FedEdgeIDS is an operationally feasible, privacy-preserving, and scalable solution for next-generation network security in IoT-intensive environments.

Future research directions building upon this work include:

- **State-of-the-Art Deep Learning Models:** Transformer-based architectures (e.g., BERT-based traffic encoders) combined with Graph Neural Networks (GNNs) to capture inter-device communication relationships. Early experiments indicate that GNN-based approaches can further reduce FPR by approximately 0.15–0.25% through detection of topological anomalies in communication graphs.
- **Differential Privacy Integration:** Integration of a formally verified R´enyi Differential Privacy accountant into the federated aggregation protocol to provide mathematically rigorous privacy guarantees on gradient information, ensuring compliance with regulatory requirements in sensitive industrial deployments.
- **Blockchain-Based Federated Trust:** Investigation of a permissioned blockchain layer (e.g., Hyperledger Fabric) to maintain an immutable audit log of model updates, enabling post-hoc forensic attribution of model poisoning attempts and cryptographically verifiable model version control.
- **Adversarial Robustness Hardening:** Evaluation of adversarial training methods such as FGSM, PGD, and AutoAttack to improve robustness against evasion attacks designed to bypass machine learning-based detectors.
- **Real-World Industrial Deployment:** Pilot deployment in collaboration with an industrial partner on an operational SCADA network to evaluate FedEdgeIDS under real-world conditions with live adversarial activity.
- **Continual Learning:** Incorporation of continual learning mechanisms to enable adaptation to evolving attack patterns without catastrophic forgetting, addressing concept drift in dynamic network environments.

FedEdgeIDS will be released as open-source software upon acceptance of this paper, including source code, trained model weights, and evaluation scripts, to support reproducibility and facilitate further research by the community.

References

1. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 119–129 (2017)
2. Chen, J., Li, X., Lai, Y., Wu, J.: Blockchain-based federated learning for intrusion detection in industrial internet of things. IEEE Transactions on Industrial Informatics **19**(2), 1601–1610 (2023)
3. Diro, A.B., Chilamkurti, N.: Distributed attack detection scheme using deep learning approach for internet of things. Future Generation Computer Systems **82**, 761–768 (2018)
4. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-iiotset: A new comprehensive realistic cybersecurity dataset. IEEE Access **10**, 40281–40306 (2022)
5. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies. pp. 21–26 (2016)
6. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity **2**(1), 1–22 (2019)
7. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Smola, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proceedings of MLSys (2020)
8. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). pp. 1273–1282 (2017)
9. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G.: A survey on security and privacy of federated learning. Future Generation Computer Systems **115**, 619–640 (2021)
10. Moustafa, N., Creech, G., Slay, J.: Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In: Data Analytics and Decision Support for Cybersecurity, pp. 127–156. Springer, Cham (2017)
11. Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R.: Diot: A federated self-learning anomaly detection system for iot. In: Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS). pp. 756–767 (2019)
12. Premsankar, G., Di Francesco, M., Taleb, T.: Edge computing for the internet of things: A case study. IEEE Internet of Things Journal **5**(2), 1275–1284 (2018)
13. Rey, V., Sa´nchez, P.M.S., Celdra´n, A.H., Bovet, G.: Federated learning for malware detection in iot devices. Computer Networks **204**, 108693 (2022)
14. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset. In: International Conference on Information Systems Security and Privacy. pp. 108–116 (2018)

15. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: Vision and challenges. IEEE Internet of Things Journal **3**(5), 637–646 (2016)
16. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6 (2009)
17. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954–21961 (2017)
18. Zhang, Q., Yang, L.T., Chen, Z., Li, P.: A survey on deep learning for big data. Information Fusion **42**, 146–157 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

