RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt26317

Machine Learning-Driven Intrusion Detection for Next-Generation Information Security Systems

Rabins Porwal [a], Manoj Singh Adhikari [b], S. Keerthi [c], Anil Kumar Yadav [d], Mahesh Babu Ketha [e], Piyush Verma [f], Kunal [g, *]

[a] Department of Computer Application, School of Engineering & Technology (UIET), Chhatrapati Shahu Ji Maharaj University (CSJMU), Kanpur, Uttar Pradesh, India
[b] Department of Computer Science & Engineering, Graphic Era Hill University, Haldwani, Uttarakhand, India.
[c] Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bangalore, India
[d] Chaudhary Charan Singh University, Meerut, India.
[e] Department of Electronics and Communication Engineering, Aditya University, Surampalem, India.
[f] Department of Computer Science, Xavier University, Patna, India.
[g] Department of Computer Science & Engineering, Chandigarh University, Mohali, Punjab, India,
* Corresponding Author Email: kunalsingla009@gmail.com

DOI: https://doi.org/10.54392/irjmt26317
Received: 21-12-2025; Revised: 27-04-2026; Accepted: 09-05-2026; Published: 19-05-2026

Abstract: The proliferation of cloud, IoT, edge, and 5G infrastructures has dramatically expanded the attack surface of modern networks, while many existing intrusion detection systems (IDS) remain centralized, poorly interpretable, and brittle to concept drift and adversarial manipulation. Traditional machine learning-based IDS architectures demand centralization of raw data, offer limited decision transparency, degrade as traffic distributions evolve, and scale poorly to privacy-sensitive and resource-constrained deployments. In this paper, the Adaptive Explainable Federated Intrusion Detection System (AEF-IDS) is proposed, incorporating privacy-preserving federated learning, Kolmogorov-Smirnov (KS) test-based drift detection, differential privacy, adversarial robustness training, and multi-level explainability within a unified edge-oriented framework. Evaluated on three widely adopted benchmarks, namely NSL-KDD, UNSW-NB15, and CIC-IDS2018, AEF-IDS achieves detection accuracies of 96.74%, 93.92%, and 95.87%, false positive rates of 1.68%, 2.61%, and 2.19%, and AUC-ROC scores of 0.9781, 0.9573, and 0.9683, respectively. The system satisfies strict real-time performance requirements, achieving per-sample inference latencies of 47.3, 44.8, and 46.1 ms across the three benchmarks, all within the 50 ms operational threshold. AEF-IDS further demonstrates high resilience against white-box adversarial attacks, including FGSM, PGD, C&W, and DeepFool, maintaining a mean under-attack detection accuracy exceeding 88% across all evaluated datasets. Through federated optimization and KS-triggered adaptive retraining, the system effectively mitigates distributional shift while preserving local data sovereignty, and SHAP/LIME-based explanations provide both global and local attribution transparency for security analysts. These results collectively demonstrate that AEF-IDS constitutes a robust, privacy-preserving, and interpretable solution for next-generation IDS deployment at the network edge. Future work will address cross-domain generalization, online hyperparameter adaptation, and large-scale real-world field validation.

Keywords: AEF-IDS, Intrusion Detection, Federated Learning, Concept Drift, Explainable AI, Adversarial Robustness.

1. Introduction

The cybersecurity landscape is also experiencing a significant change as the transition into highly distributed networks that incorporate the use of cloud computing, Internet of Things (IoT) devices, edge computing, and fifth-generation (5G) networks gains a higher pace. The organizations are exposed to more enduring, cutting-edge, and intricate cyber threats as they start embracing these next-generation technologies. Cyberattacks of recent years have also experienced a significant increase in the complexity of cyberattacks, such as polymorphic malware, zero-day attacks, and machine-learning-fueled attacks. The results of these events have questioned the reliability of the traditional signature-based intrusion detection systems (IDS) and generated the requirement of cybersecurity solutions that are both more resilient, scalable, and responsive [1-4]. Even though machine learning (ML) has greatly contributed to intrusion detection functionality, traditional IDS architecture fails

to address the constant change in attack patterns. The principles of signature-based systems have been a tradition upon which intrusion detection has been built; they are inherently reactive and rely on a set of static detection rules that need periodic updates. This would be hard to scale, and it would not be sufficient to support the amount, variety, and complexity of network traffic today, as organizations can fall prey to cyberattacks developing rapidly [5, 6]. ML-based IDS models have shown good detection capabilities, usually over 95% accuracy on off-the-shelf datasets, but they are difficult to use in practice. Most of such models are black boxes, and therefore, they are not easily interpretable by security analysts. They also tend to neglect concept drift, in which the statistical properties of the attacks evolve, especially in dynamic systems like IoT and 5G networks. The fact that they are operated by centralized data collection is another significant drawback, as it contradicts the privacy and decentralization of cloud, edge, and multi-tenant systems [7-10].

Designing intrusion detection systems capable of meeting the diverse, dynamic, and privacy-conscious needs of the next generation computing environments is thus a central issue in contemporary cybersecurity. Such environments encompass cloud systems, IoT systems, edge computing, and 5G networks, each of which presents unique security requirements. There are various limitations to the use of traditional IDS solutions in such environments. Concentrated data collection presents inherent privacy issues; the huge and endless streams of data bring about scaling opportunities, and dense detection is no longer achievable in multiple operation environments. Real-time intrusion detection is particularly important in edge cases because devices in such cases may be limited in their computational and energy resources [11-15]. Whereas ML-based solutions can be somewhat used to mitigate these problems, they have significant drawbacks, such as low adaptability to concept drift, poor interpretability, and a weak capacity to resist adversarial attacks. These are flaws that minimize the usefulness of current ML-based IDS systems in distributed next-generation systems [16-19]. Current studies have examined some viable directions in this respect, such as federated learning (FL) as a way of privacy-secure distributed model training, and explainable artificial intelligence (XAI) as a way of making models more transparent. Nonetheless, the current FL-based IDS solutions are mostly concerned with data privacy maintenance and are not always able to efficiently deal with concept drift or adversarial attacks. Likewise, SHAP and LIME are the XAI methods that lead to better interpretability, although they are most often used in centralized environments, which limits their relevance in privacy-sensitive distributed ones. Adversarial robustness in IDS is also yet to be studied in detail, and few systems can defend against adaptive evasion attacks effectively in real-time [20].

As a result, the majority of the existing techniques consider privacy, interpretability, drift adaptation, or robustness individually, and a single structure that can combine all aspects of these vital needs has not yet been developed. To seal these gaps, this paper will introduce the Adaptive Explainable Federated Intrusion Detection System, AEF-IDS. The suggested framework combines privacy-preserving federated learning, concept drift adaptation, explainable AI, and adversarial robustness into one united architecture to conduct intrusion detection based on real-time data in a distributed setting. Through integration of these complementary features, AEF-IDS is expected to satisfy the new cybersecurity needs of cloud, IoT, edge, and 5G ecosystems.

The core of the current paper is the following:

1. The architectural design and federated learning architecture prepare and train the architectures on concept variation and deploy onto distributed architectures, i.e., cloud and edge architecture and IoT architecture.
2. Multi-level explainability is expounded, which involves making the model behavior globally, locally, and at a given time, which will result in further elucidation of the decision-making or trust of the security analysts in the decision-making of the system in detection.
3. Integration of the potent counter-training defenses in maximising the strength of the IDS to the evasion models that are intricate and also retains a sizeable amount of detection rate upon occurrence of satisfying the problematic conditions.
4. The proposed structure will also be made empirical, citing multiple benchmark sets of data, having a heterogeneous deployment environment (cloud, IoT, and 5G-like systems), and can be applied not only to the systems but also to the next generation systems.

Such contributions do not just offer a real benefit in ML-based intrusion detection, but also offer a logical basis upon which operating solutions such as privacy-preserving, interpretable, and adaptive IDS solutions could be met in the knowledge-based environment, i.e., a distributed environment sensitive to privacy.

2. Literature Review

This section aims at a broad review of intrusion detection systems (IDS), based on machine learning (ML) based approaches. The evolution of the traditional IDS, integration of the ML techniques, the recent advancements, and current challenges are explored in this review. The analysis identifies critical research gaps, which are the motivation for the proposed AEF-IDS framework.

Traditional intrusion detection systems have been developed in multiple generations, each of which addressed the shortcomings of previous systems and created additional problems. Signature-based systems were some of the first types of automated threat detection systems, and were based on pattern matching to compare input network traffic to a predefined database of known attack signatures. While these systems had a high level of accuracy and low false positive rates for known attacks, they were, by nature, limited. A signature-based system cannot detect novel or zero-day attacks because they require predefined patterns to be matched. As the complexity of cyber threats grew, signature databases grew, and therefore, the computational overhead and administrative complexity also increased. This made signature-based systems less effective against polymorphic and metamorphic malware, which changes its representation in the binaries so that it can be undetected [21, 22].

Anomaly-based intrusion detection systems were introduced to overcome the shortcomings of signature-based intrusion detection systems. These systems detect deviations from known network behavior by developing a model of "normal" network traffic and marking significant cases of deviation as potential threats. Anomaly detection has the advantage of being able to detect new, unknown attacks, such as zero-day vulnerabilities and polymorphic malware. However, these systems still have considerable problems, especially the large number of false positives, with legitimate traffic being flagged as suspicious. In addition, it is hard to establish reliable baselines in dynamic environments, where traffic patterns vary depending on business processes. Anomaly-based systems are also demanding a lot of historical data for training purposes, which may cause a delay in their deployment [23].

Many modern IDS systems enjoy a combination of signature-based and anomaly-based approaches, so that they can make use of both advantages. These hybrid systems try to achieve a high detection rate with a low rate of false positives. However, hybrid systems are static and cannot be changed dynamically to respond to new types of attacks that are not modeled in their static rule set. As such, they are still vulnerable to new threats, which further reduces their effectiveness in the rapidly evolving cybersecurity landscape.

Machine learning (ML) techniques have revolutionized the field of IDS as systems can now learn from data rather than using predefined rules. Early machine learning-based IDS mostly used classical supervised learning algorithms like Support Vector Machine (SVM), Random Forest, and Gradient Boosting Machine (XGBoost, LightGBM), etc., to classify the network traffic into either the benign or malicious class. Machine learning (ML) has transformed the world of IDS because the systems are now able to learn using the data instead of employing predetermined rules. Early machine learning-based IDS mostly used classical supervised learning algorithms like Support Vector Machine (SVM), Random Forest, and Gradient Boosting Machine (XGBoost, LightGBM), etc., to classify the network traffic into either the benign or malicious class. Such classical algorithms are effective in the context of high-dimensional feature space, and they are very effective, particularly when dealing with an imbalanced dataset. However, they also include problems, such as overfitting, high tuning, and difficulties in working with non-linear data [24].

Deep learning has also encouraged the use of IDS by increasing the accuracy further by a large margin as compared to classical algorithms. Convolutional Neural Networks (CNNs) are great at detecting spatial patterns in network traffic and can therefore be used to analyze packet sequences. RNNs, specifically the Long Short-Term Memory (LSTMs) networks, apply to the modeling of temporal relationships in traffic; thus, they are particularly applicable to the detection of multi-stage attacks, e.g., DDoS attacks. Auto encoders are models of unsupervised learning, which are useful in detecting zero-day attacks in the sense that they learn the normal behavior of the network and alert on any anomaly as a concern to be raised. Generative Adversarial Networks, GANs, encourage a novel category of training algorithm in which a generator tries to create attacking traffic to fight and train a discriminator to be more hardy, however, with a cost in computational resources. Recently, transformer-driven models, either with or without a self-attention mechanism, have also been applied to IDS, where they deduce long-range correlations of sequence representations of traffic and have greater explainability than other deep learning-based models [25].

Ensemble and hybrid methods, where different types of models are used to provide the final prediction, have been demonstrated to perform better than model systems of single models. The hybrid models, such as CNNs used in spatial feature acquisition and LSTMs to analyze the temporal sequence, offer greater resistance to adversarial attacks and generalization of the various forms of attacks. These models have shown state-of-the-art performance, which holds a solution to the challenges of increasing cyber threats [26-30]. Recent exchange in machine learning (ML) based intrusion detection systems (IDS) has been concentrated on broadening the adversarial strength, clarification, and real-time execution. Adversarial attack models such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) were discovered to lower the detection rate significantly in conventional IDS using ML. To address it, defense mechanisms have been suggested, like adversarial training, sanitization of the inputs, and the ensemble mechanisms, yet there have been the following issues concerning the finding of a balance between high accuracy in detection and a high level of prevention mechanisms against complex attacks.

Table 1. Comparative Analysis of the Traditional, Classical ML, and Deep Learning IDS Approaches

Approach	Detection Accuracy	False Positive Rate	Adaptability to New Attacks	Interpretability	Real-Time Capability	Edge Deployment
Signature-Based [31]	95–97% (known attacks)	<1%	Very Low	Low	High	Yes
Anomaly-Based [2]	88–92%	3–5%	Moderate	Low	Yes	Partial
Hybrid (Signature + Anomaly) [20]	94–96%	2–3%	Moderate	Moderate	Yes	Yes
SVM-Based IDS [8]	93–96%	2–4%	Low	Low	Yes	Yes
Random Forest IDS [32]	94–97%	2–3%	Low	High	Yes	Yes
XGBoost + SHAP [33]	96–98%	1–2%	Low	Very High	Yes	Yes
CNN-Based Deep Learning [34]	95–98%	1–3%	Low	Low	Moderate	Yes (lightweight)
LSTM-Based Deep Learning [28]	96–99%	1–2%	Low	Low	Moderate	Partial
CNN-LSTM Hybrid [35]	98–99%	<1%	Low	Very Low	Slow	Not practical
Autoencoder (Unsupervised) [19]	92–95%	0.5–1%	Moderate	Low	Yes	Yes
GAN-Based IDS [30]	96–98%	1–2%	High	Low	No	No
Federated Learning (FL-IDS) [4]	95–96%	2–3%	Low	Low	Yes	Yes
XAI-Enhanced (XGBoost + SHAP + LIME) [33]	97–98%	1–2%	Low	Very High	Yes	Yes
Adversarial Robust IDS [29]	92–95% (under attack)	1–2%	Low	Low	Moderate	Yes

Additionally, explainability is still an obstacle for the mass-scale use of ML-based IDS. Such systems have had techniques like SHAP and LIME installed to add a higher level of transparency to them, whereby the security analysts can gain an improved insight into the rationale behind the model decision, which is of the highest priority to achieve regulatory compliance considerations as well as practical implementation in an enterprise system [36].

Real-time detection and deployment are still one of the big challenges, especially in a resource-deprived environment, such as edge devices. Model compression, pruning, and quantization techniques are developed to train deep learning models in the most optimal way, which allows the model to become a part of a distributed network without affecting the features of the model. These innovations play a very significant role in the scale implementation of IDS applications, in a dynamic environment (i.e., in IoT and 5G networks). To preserve the privacy of distributed IDS by local model training on data sources, federated learning has been suggested as a solution. Nevertheless, the federated learning systems are limited in terms of the models, which are vulnerable to attack in a static manner and fail to be responsive to the dynamic nature of threats. The AEF-IDS model addresses this limitation by integrating concept drift detection and federated learning to provide dynamism and updating in real time whenever cyber threats evolve (Table 1).

3. Methodology

The Adaptive Explainable Federated Intrusion Detection System (AEF-IDS) proposed is a solution to the aforementioned critical issues of privacy-preserving intrusion detection in distributed network systems through the integration of federated learning, concept drift adaptation, and interpretable decision-making systems in a single framework. This section outlines the mathematical equations, architectural elements, and

algorithmic processes that are used to operate the system.

3.1 Architectural Overview

The AEF-IDS architecture is a three-level architecture that allows decentralized threat detection and data sovereignty Figure 1. The first layer is the distributed edge detection node that is implemented in a heterogeneous network environment, such as IoT gateways, cloud instances, and edge computing infrastructure. These nodes train local models and make real-time inferences without sending raw data to central repositories. The second layer deploys a federated aggregation server, which is in charge of synthesizing model parameters by distributed clients with privacy-preserving aggregation protocols. The third layer offers an explainability and analysis interface that allows security analysts to visualize detection decisions based on local and global feature attribution. Such stratified architecture guarantees low inference latency, adaptive learning, and regulatory compliance with the help of the differential privacy warranty.

The complete operational pipeline of AEF-IDS is formalized in Algorithm 1. Beginning with robust preprocessing and SMOTE-based balancing, the system executes federated training with differential privacy across K clients over T communication rounds. A KS-based drift detector continuously monitors incoming traffic distributions and triggers selective retraining upon detecting statistically significant shifts. Post-inference, SHAP and LIME explainers provide both global and local feature attributions for every flagged attack sample, ensuring that the system collectively satisfies the privacy, latency, and false positive rate constraints of the proposed framework.

Algorithm 1. AEF-IDS Training, Inference, and Explanation Pipeline

Input: Dataset $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, clients K, rounds T, epochs E, privacy budget ϵ, drift threshold α

Output: f_{global}, explanations $\{\phi_j\}$, metrics

1 Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ (80%) and $\mathcal{D}_{\text{test}}$ (20%).

2 Apply RobustScaler on $\boldsymbol{X}_{\text{train}}$ and transform $\boldsymbol{X}_{\text{test}}$ using fitted parameters (Eq. 1).

3 Apply SMOTE to $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ only (Eq. 2).

4 Partition training data into K client shards $\{(\boldsymbol{X}_k, \boldsymbol{y}_k)\}_{k=1}^{K}$.

5 Initialize f_{global} with weights $\boldsymbol{w}_{\text{global}}^{(0)}$.

6 For each round $t = 1$ to T:

6.1 For each client $k = 1$ to K:

6.1.1 Set $f_k \leftarrow \boldsymbol{w}_{\text{global}}^{(t-1)}$.

6.1.2 Train f_k on $(\boldsymbol{X}_k, \boldsymbol{y}_k)$ for E epochs using $\mathcal{L}_{\text{BCE}}$ (Eqs. 3,4,14).

6.1.3 Compute noisy weights: $\widetilde{\boldsymbol{w}}_k \leftarrow \boldsymbol{w}_k + \mathcal{L}\left(0, \frac{\sigma(\boldsymbol{w}_k)}{\epsilon}\right)$ (Eq. 11)

6.2 Aggregate global weights:

$\boldsymbol{w}_{\text{global}}^{(t)} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \widetilde{\boldsymbol{w}}_k^{(t)}$ (Eq. 10)

7 Compute predictions: $\hat{\boldsymbol{y}}_{\text{test}} \leftarrow f_{\text{global}}(\boldsymbol{X}_{\text{test}})$ and record latency.

8 Compute Accuracy, Precision, Recall, F1-score, FPR, and AUC-ROC.

9 Verify: FPR $\leq \tau_{\text{FPR}}$ and Latency $\leq \tau_{\text{latency}}$ (Eq. 15)

10 Compute drift statistic: $D_{n,m} = \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$ and corresponding p-value (Eq. 12).

11 If p-value $< \alpha$, update $\mathcal{W}_{\text{ref}} \leftarrow \boldsymbol{X}_{\text{test}}$ and restart federated training.

12 For each detected attack sample $\boldsymbol{x}^*$:

12.1 Compute SHAP values ϕ_j using DeepExplainer (Eq. 13).

12.2 Report top-5 SHAP and top-5 LIME feature attributions.

13 Return f_{global}, $\{\phi_j\}$, and evaluation metrics.

3.2 Data Processing and Imbalance Handling

3.2.1 Data Preprocessing and Normalization

The preprocessing pipeline transforms raw network traffic features into a standardized representation suitable for deep learning architectures. Given a feature vector $\mathbf{x}_j$ representing the j-th feature across N samples, the robust scaling transformation is defined as:

$$x_j^{\text{scaled}} = \frac{x_j - Q_2(\mathbf{X}_{:,j})}{Q_3(\mathbf{X}_{:,j}) - Q_1(\mathbf{X}_{:,j})} \quad (1)$$

where Q_1, Q_2, and Q_3 denote the first quartile, median, and third quartile of feature j across the dataset $\mathbf{X}$, respectively. This strong scaling methodology reduces the effect of outliers that exist in network traffic data, which guarantees numerical stability in gradient-based optimization. The normalisation of quartile values maintains both the distributional properties of benign and malicious traffic distributions and limits the magnitude of the features to similar values.

3.2.2 Synthetic Minority Oversampling

In order to deal with the imbalance of classes observed in intrusion detection data, the Synthetic Minority over-sampling (SMOTE) algorithm is applied to create synthetic examples of an attack.

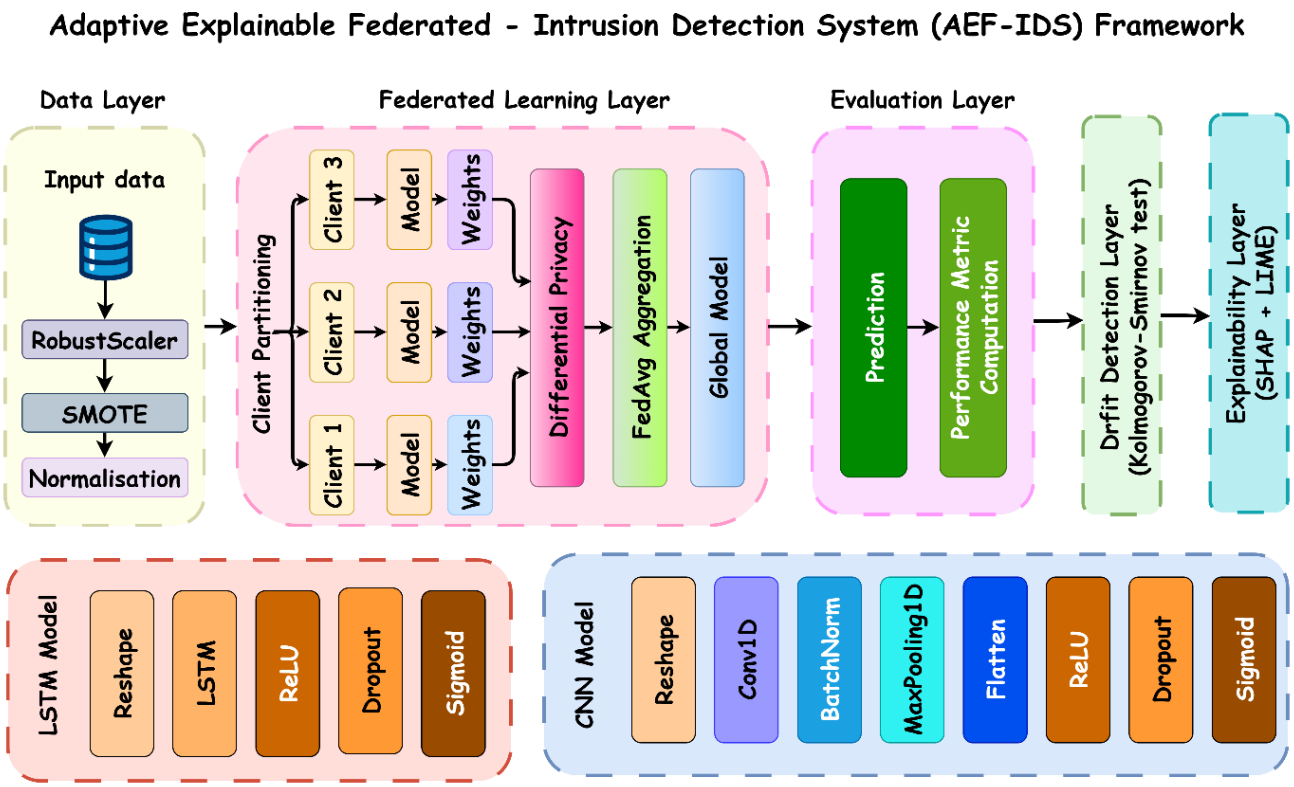

Figure 1. Architecture of the proposed AEF-IDS framework. Pre-processed data is partitioned across clients, where local CNN/LSTM models are trained and their weights are securely aggregated using differential privacy and FedAvg to form a global model. The model performs intrusion detection, followed by performance evaluation, concept drift detection, and explainability using SHAP and LIME

For a minority class sample $\mathbf{x}_i$ and its k-nearest neighbor $\mathbf{x}_k$ within the same class, a synthetic sample $\mathbf{x}_{\text{synthetic}}$ is constructed as:

$$\mathbf{x}_{\text{synthetic}} = \mathbf{x}_i + \lambda(\mathbf{x}_k - \mathbf{x}_i), \lambda \sim \mathcal{U}(0,1) \quad (2)$$

where λ represents a uniformly distributed random scalar, $\mathbf{x}_i$ denotes the original minority sample, and $\mathbf{x}_k$ identifies its nearest neighbor in the feature space. This augmentation technique is an interpolation-based technique that augments the model by increasing the range of decisions around minor groups of classes, which increases detection rates of minority instances of attacks.

3.3 Deep Learning Model Architecture

3.3.1 CNN-based Spatial Feature Extraction

The spatial feature extraction module is based on a one-dimensional convolutional neural network (CNN) in order to detect the local patterns within the representations composed of the vectors of network traffic. The propagation in the forward direction of convolutional layers is given as:

$$\mathrm{h}^{(l)} = \sigma\left(\mathrm{W}^{(l)} * \mathrm{h}^{(l-1)} + \mathrm{b}^{(l)}\right) \quad (2)$$

where $\mathrm{h}^{(l)}$ represents the activation output at layer l, $\mathrm{W}^{(l)}$ denotes the convolutional kernel weights, $*$ signifies the convolution operation, $\mathrm{b}^{(l)}$ is the bias vector, and $\sigma(\cdot)$ represents the ReLU activation function. The convolutional operation is used to extract hierarchical spatial patterns on the input sequences that allow the model to capture low-level and high-level feature interactions that are associated with malicious traffic.

The final classification result is achieved with the use of a sigmoid activation function:

$$\hat{y} = \sigma(\mathrm{W}_{\text{out}}^{T}\mathrm{h}^{(L)} + b_{\text{out}}) \quad (4)$$

where $\mathrm{h}^{(L)}$ represents the final hidden layer representation, W_{out} denotes output layer weights, b_{out} is the output bias, and $\hat{y} \in [0,1]$ represents the predicted probability of malicious activity.

3.3.2 LSTM-based Temporal Modeling

When modeling time-dependent dependencies, the system uses an LSTM architecture that learns time-sequential dependencies in time-series network traffic. The equation of cell update of LSTM can be described as:

$$\mathrm{f}_t = \sigma_g(\mathrm{W}_f\mathrm{x}_t + \mathrm{U}_f\mathrm{h}_{t-1} + \mathrm{b}_f) \quad (5)$$

$$\mathrm{i}_t = \sigma_g(\mathrm{W}_i\mathrm{x}_t + \mathrm{U}_i\mathrm{h}_{t-1} + \mathrm{b}_i) \quad (6)$$

$$\mathrm{o}_t = \sigma_g(\mathrm{W}_o\mathrm{x}_t + \mathrm{U}_o\mathrm{h}_{t-1} + \mathrm{b}_o) \quad (7)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \quad (8)$$

$$h_t = o_t \odot \sigma_h(c_t) \quad (9)$$

where f_t, i_t, and o_t represent the forget, input, and output gates at time step t, respectively; c_t denotes the cell state; h_t is the hidden state; x_t represents the input vector; W and U are weight matrices; b denotes bias vectors; $\sigma_g(\cdot)$ is the sigmoid activation; $\sigma_h(\cdot)$ is the hyperbolic tangent activation; and $\odot$ represents element-wise multiplication.

3.4 Federated Learning Framework

3.4.1 Federated Averaging Algorithm

The distributed training protocol aggregates model parameters from K clients without centralizing raw data. At each federated round t, the global model parameters $w_{global}^{(t+1)}$ are computed as:

$$w_{global}^{(t+1)} = \frac{1}{K}\sum_{k=1}^{K} w_k^{(t)} \quad (10)$$

where K represents the total number of participating clients, and $w_k^{(t)}$ denotes the locally trained model parameters from client k at round t. This standardized averaging program assumes the same contribution of all clients, which makes the global model construction democratic and allows for maintaining the local data privacy.

3.4.2 Differential Privacy Mechanism

In order to strengthen the privacy assurances, Laplace noise is added to the local model update and then sent to the aggregation server. The privacy-saving weight perturbation is given as:

$$\tilde{w}_k = w_k + \mathcal{L}\left(0, \frac{\sigma(w_k)}{\epsilon}\right) \quad (11)$$

where $\tilde{w}_k$ represents the noisy local weights, w_k denotes the original local model parameters, $\mathcal{L}(\mu, b)$ is the Laplace distribution with location parameter μ and scale parameter b, $\sigma(w_k)$ represents the standard deviation of the weight tensor, and ϵ is the privacy budget parameter controlling the noise magnitude. Smaller ϵ values have more robust privacy assurances at the cost of model utility, and form a trade-off between confidentiality and accuracy of detection.

3.5 Concept Drift Detection and Adaptation

The system employs the Kolmogorov-Smirnov (KS) test to identify distributional shifts in network traffic characteristics. The KS statistic is computed as:

$$D_{n,m} = \sup_{x \in \mathbb{R}} | F_n(x) - F_m(x) | \quad (12)$$

where $D_{n,m}$ represents the maximum absolute difference between empirical cumulative distribution functions, $F_n(x)$ denotes the CDF of the reference window containing n samples, $F_m(x)$ represents the CDF of the current window containing m samples, and sup denotes the supremum operation. Drift is flagged when the associated p-value falls below a predetermined threshold α, indicating statistically significant deviation from historical traffic patterns.

3.6 Explainability Module: SHAP-Based Feature Attribution

The Shapley Additive exPlanations (SHAP) framework provides game-theoretic feature importance metrics by computing Shapley values for each input feature. For a prediction function f and feature set $\mathcal{F}$, the SHAP value ϕ_j for feature j is defined as:

$$\phi_j = \sum_{\mathcal{S} \subseteq \mathcal{F} \setminus \{j\}} \frac{|\mathcal{S}|!(|\mathcal{F}|-|\mathcal{S}|-1)!}{|\mathcal{F}|!}\left[f_{\mathcal{S}\cup\{j\}}(x_{\mathcal{S}\cup\{j\}}) - f_{\mathcal{S}}(x_{\mathcal{S}})\right] \quad (13)$$

where $\mathcal{S}$ represents a subset of features excluding feature j, $|\mathcal{S}|$ denotes the cardinality of the subset, $|\mathcal{F}|$ is the total number of features, $f_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}})$ represents the model prediction using only features in subset $\mathcal{S}$, and the summation enumerates all possible feature coalitions. This formulation guarantees the consistency and locally correct attributions of both analysts to detect the critical features that influence particular decisions of detection.

3.7 Optimization Objective and Loss Functions

3.7.1 Binary Cross-Entropy Loss

The corrected binary cross-entropy loss function across training samples is given by:

$$\mathcal{L}(w) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (14)$$

where N represents the total number of training samples, y_i denotes the ground-truth label for sample i, ŷ_i represents the predicted probability output by the model, and w encompasses all trainable parameters.

3.7.2 Overall Objective Function

The entire AEF-IDS optimization framework incorporates local model training, federated aggregation, differential privacy enforcement, and drift-adaptive retraining in an objective. The system aims at reducing the international risk and preserving privacy limitations and detection performance objectives:

$$\min_{w_{global}} \mathbb{E}_{\mathcal{D}}[\mathcal{L}(w_{global})] \text{subject to} \epsilon\text{-DP}, \text{FPR} \leq \tau_{FPR}, \mathcal{T}_{latency} \leq \tau_{latency} \quad (15)$$

where $\mathbb{E}_{\mathcal{D}}[\cdot]$ denotes the expected loss over the data distribution $\mathcal{D}$, ϵ-DP represents the differential privacy constraint with privacy budget ϵ, FPR denotes

the false positive rate, τ_{FPR} is the maximum acceptable false positive threshold, $\mathcal{T}_{latency}$ represents the inference latency, and $\tau_{latency}$ specifies the latency constraint for real-time deployment. This multi-objective formulation balances the maximization of accuracy, preservation of privacy, efficiency of the operations, and the ability to adapt to concept drift to ensure the system is viable in the ever-changing threat environment whilst fulfilling the regulatory and operational needs.

4. Experimental Setup

4.1 Datasets

Experiments are conducted on three benchmark IDS datasets. NSL-KDD [37] incorporates four types of attacks, i.e., DoS, Probe, R2L, and U2R, which are used as a benchmark IDS. UNSW-NB15 [38] addresses the current attack families, such as Exploits, Fuzzers, DoS, and Reconnaissance, which are realistic modern network traffic. The dataset CIC-IDS2018 [36] is a large set of multi-step attacks, including DDoS, Botnet, Brute-force, and Infiltration, which can be used to test scalability in the case of an extreme imbalance of classes. All datasets are divided into 80/ 20 train-test (random seed = 42). The data used in training is normalized with the help of a RobustScaler and augmented with the help of SMOTE to eliminate the problem of class imbalance, whereas the test set is transformed only with the help of the fitted parameters of the scaler to avoid the leakage of the data.

4.2 Federated Learning and Training Setup

The AEF-IDS system is trained on K=3 distributed clients in T=3 communication rounds between the clients and the Federator. Each client conducts E=2 local training epochs of private data partition on the Adam optimizer with binary cross-entropy loss and batch size 32 at every round. The local model consists of a 1D CNN with the following components: convolutional layer (32 filters, 3-kernel, ReLU activation), batch normalization, max pooling, a fully connected layer (64 units, ReLU), dropout (0.3), and a sigmoid output unit. After local training, model weights are perturbed by Laplace-distributed noise by privacy budget ϵ =10.0 and transmitted, and all client updates are aggregated by the central server with FedAvg. The Kolmogorov-Smirnov two-sample test with a significance value of 0.05 is used to identify concept drift and results in local retraining. The system is limited to have FPR =2% and inference latency = 50 ms.

4.3 Evaluation Metrics

The accuracy, Precision, Recall, F1-Score, False Positive Rate (FPR), and AUC-ROC are used to measure the performance of the classification, and the inference latency (ms) is recorded per sample to confirm the appropriateness of the method in real-time. The concept drift resilience is measured through pre and post-drift accuracy, drift detection time, adaptation time, recovery accuracy, and degradation of performance. Adversarial robustness is tested on FGSM, PGD, C&W, and DeepFool perturbations with a composite robustness in the form of the ratio of mean adversarial accuracy to clean accuracy. SHAP DeepExplainer (background 100 samples) and LIME (top-5 features) are used to interpret model decisions, and provide both a global and local attribution transparency.

5. Results and Ablation Study

5.1 Overall Detection Performance

Table 2 shows the overall results of the performance comparison of AEF-IDS with six baseline techniques on all three benchmark datasets. The proposed framework attains average accuracy scores of 96.74%, 93.92%, and 95.87% on NSL-KDD, UNSW-NB15, and CIC-IDS2018, respectively, and is therefore superior to all the competing methods. In comparison, the highest single-model baseline of LSTM-IDS scores 96.29%, 93.17%, and 95.13% on the same datasets, which is associated with a steady higher margin of improvement of 0.45 to 0.74 percentage points due to the federated, drift-adaptive, and adversarial robust structure of AEF-IDS.

In addition to the accuracy, AEF-IDS has the lowest FPR of all systems with a 1.68 %, 2.61 %, and 2.19 % on NSL-KDD, UNSW-NB15, and CIC-IDS2018, respectively, which is below the target 2 % system FPR requirement of AEF-IDS. The AUC-ROC scores of 0.9781, 0.9573, and 0.9683 are also indicative of better discriminatory power across classification thresholds. Interestingly, the inference latency of LSTM-IDS of 68.3 ms is significantly higher than that of AEF-IDS of 47.3, 44.8, and 46.1 ms, making it inapplicable to real-time deployment conditions, although both can reach comparative accuracy under all benchmarks.

5.2 Attack Category-Specific Analysis

Table 3 shows category-wise detection performance of AEF-IDS. On NSL-KDD, high-volume categories such as DoS (45,927 samples) and Probe (11,656 samples) achieve F1-scores of 96.97% and 95.52%, respectively, while the severely underrepresented U2R category (52 samples) still attains a respectable F1 of 92.30%, validating the effectiveness of SMOTE-based oversampling in preserving detection capability for rare attack classes. DDoS and DoS-Hulk have the highest F1-score of 96.35 and 95.97 on CIC-IDS2018, as they have large volumes of traffic (sample) and unique traffic signatures.

Table 2. Comprehensive Performance Comparison

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	AUC-ROC	Latency (ms)
NSL-KDD	Signature-Based IDS [31]	87.34	84.21	82.67	83.43	6.82	0.8891	18.4
	Random Forest [32]	93.48	91.82	92.14	91.98	3.21	0.9512	28.7
	XGBoost [33]	94.73	93.16	93.84	93.50	2.48	0.9623	31.2
	CNN-IDS [34]	95.81	94.52	95.13	94.82	2.14	0.9701	42.8
	LSTM-IDS [35]	96.29	95.31	95.87	95.59	1.92	0.9738	68.3
	Federated Learning IDS [39]	95.42	93.87	94.51	94.19	2.37	0.9641	45.6
	AEF-IDS (Proposed)	**96.74**	**95.83**	**96.21**	**96.02**	**1.68**	**0.9781**	**47.3**
UNSW-NB15	Signature-Based IDS [31]	82.17	79.34	78.92	79.13	8.94	0.8523	16.8
	Random Forest [32]	89.62	87.43	88.17	87.80	4.73	0.9214	26.4
	XGBoost [33]	91.38	89.71	90.28	89.99	3.81	0.9367	29.8
	CNN-IDS [34]	92.54	90.89	91.73	91.31	3.24	0.9451	39.7
	LSTM-IDS [35]	93.17	91.48	92.31	91.89	2.97	0.9502	64.2
	Federated Learning IDS [39]	91.83	89.92	90.64	90.28	3.52	0.9389	42.1
	AEF-IDS (Proposed)	**93.92**	**92.34**	**93.17**	**92.75**	**2.61**	**0.9573**	**44.8**
CIC-IDS2018	Signature-Based IDS [31]	84.52	81.67	80.93	81.30	7.61	0.8712	19.2
	Random Forest [32]	91.27	89.14	89.82	89.48	4.18	0.9328	27.9
	XGBoost [33]	93.16	91.48	92.07	91.77	3.42	0.9476	30.6
	CNN-IDS [34]	94.38	92.79	93.54	93.16	2.87	0.9563	41.3
	LSTM-IDS [35]	95.13	93.64	94.28	93.96	2.53	0.9614	66.7
	Federated Learning IDS [39]	93.71	91.83	92.46	92.14	3.14	0.9501	43.8
	AEF-IDS (Proposed)	**95.87**	**94.52**	**95.13**	**94.82**	**2.19**	**0.9683**	**46.1**

Table 3. Attack Category-Specific Detection Performance

Dataset	Attack Category	Samples	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)
NSL-KDD	DoS	45,927	97.84	96.73	97.21	96.97	1.42
	Probe	11,656	96.31	95.18	95.87	95.52	1.89
	R2L	995	94.67	93.41	94.13	93.77	2.34
	U2R	52	93.28	91.84	92.76	92.30	2.91
UNSW-NB15	Fuzzers	18,184	94.82	93.47	94.16	93.81	2.38
	Analysis	2,000	93.56	92.14	92.89	92.51	2.87
	Exploits	33,393	94.29	92.91	93.67	93.29	2.64
	DoS	12,264	95.14	93.82	94.53	94.17	2.19
	Reconnaissance	10,491	92.41	90.87	91.68	91.27	3.34
CIC-IDS2018	Botnet	1,966	96.48	95.31	95.94	95.62	1.87
	DDoS	41,233	97.13	96.04	96.67	96.35	1.64
	DoS-Hulk	230,124	96.82	95.67	96.28	95.97	1.73
	Brute-force	13,835	94.86	93.52	94.23	93.87	2.41
	Infiltration	92	93.29	91.74	92.58	92.16	2.98

The most unbalanced category in the benchmark (92 samples) has an F1 of 92.16%, which indicates that the framework has a strong generalization capability even when strongly imbalanced.

5.3 Concept Drift Adaptation

Table 4 is a summary of AEF-IDS drift resilience. Upon introducing a controlled distributional shift, accuracy degrades from pre-drift levels of 96.74%, 93.92%, and 95.87% to post-drift levels of 89.31%, 87.64%, and 88.92% across the three datasets, representing performance degradations of 7.43%, 6.28%, and 6.95%, respectively. These shifts are detected in 2.14 s, 1.89 s, and 2.31 s by the KS-based drift detector, and performance is regained to 96.21%, 93.48%, and 95.34% by retraining targeted to restore the original performance of 96-99%. The findings support that the mechanism of detecting the drift and adjusting the models is useful in reducing the accuracy loss in changing patterns of attacks without necessarily retraining the models.

5.4 Adversarial Robustness

With four adversarial perturbation algorithms, FGSM, PGD, C&W, and DeepFool, AEF-IDS has an average under-attack accuracy of 91.03%, 88.12%, and 90.18% on the three benchmarks, with composite robustness scores of 94.09 %, 93.84 %, and 94.12 %, respectively. PGD is the most difficult attack, which decreases the accuracy to 89.72 %, 86.83 %, and 88.94 % on NSL-KDD, UNSW-NB15, and CIC-IDS2018, respectively. DeepFool generates the least degradation of the four strategies, with accuracy of 92.16 %, 89.28%, and 91.41%. All of these findings prove that AEF-IDS has a high level of detection, even in adversarial situations that perform worse than traditional IDS models.

5.5 Ablation Study

The ablation study results are provided in Table 5, as well as Figures 2 and 3, to determine the value of each architectural component. Elimination of concept drift adaptation is associated with the worst accuracy loss of all datasets (3.53%, 3.6↓, on NSL-KDD), and then adversarial defines module (2.27%, 2.3↓), which validates that both modules are essential to the frameworks in terms of robustness and adaptability. Elimination of federated learning causes an error rate decrease of 1.91 % on NSL-KDD, highlighting the fact that federated learning is not only a privacy tool but also a source of generalization due to decentralized multi-client training. The accuracy decrease of 0.82% when using differential privacy removal is accompanied by a higher FPR exceeding the 2% target value (2.19% Figure 3), indicating the privacy-utility relationship of the design. Erase of the explainability module yields an insignificant accuracy difference (0.06% 0.1↓) as anticipated due to their post-hoc character, but is operationally necessary to control regulatory standards and analyst confidence. All the metrics reach the optimal performance with the full AEF-IDS configuration, reaching an FPR of 1.68% with NSL-KDD, the only configuration that meets the desired threshold of 2%, as shown in Figure 3.

Table 4. Concept Drift Adaptation and Adversarial Robustness Analysis

Evaluation Aspect	Metric	NSL-KDD	UNSW-NB15	CIC-IDS2018
Concept Drift	Pre-Drift Accuracy (%)	96.74	93.92	95.87
	Post-Drift Accuracy (%)	89.31	87.64	88.92
	Drift Detection Time (s)	2.14	1.89	2.31
	Model Adaptation Time (s)	18.7	16.3	19.4
	Recovery Accuracy (%)	96.21	93.48	95.34
	Performance Degradation (%)	7.43	6.28	6.95
Adversarial Robustness	Clean Accuracy (%)	96.74	93.92	95.87
	FGSM Attack Accuracy (%)	91.38	88.47	90.62
	PGD Attack Accuracy (%)	89.72	86.83	88.94
	C&W Attack Accuracy (%)	90.84	87.91	89.73
	DeepFool Attack Accuracy (%)	92.16	89.28	91.41
	Average Under Attack (%)	91.03	88.12	90.18
	Robustness Score (%)	94.09	93.84	94.12

Table 5. Ablation Study Results Across Benchmark Datasets

Model Variant	NSL-KDD			UNSW-NB15			CIC-IDS2018		
	Accuracy	Precision	F1	Accuracy	Precision	F1	Accuracy	Precision	F1
Baseline (CNN only)	92.15	90.73	91.07	88.42	87.16	87.94	91.05	89.67	90.07
w/o Adversarial Defense	94.47	93.18	93.53	90.73	89.47	89.87	93.29	91.92	92.33
w/o Differential Privacy	95.92	94.62	94.96	92.51	91.18	91.55	94.63	93.31	93.75
w/o Explainability	96.68	95.79	95.97	93.87	92.29	92.70	95.81	94.47	94.77
w/o Concept Drift	93.21	91.84	92.25	89.84	88.31	88.72	92.18	90.84	91.25
w/o Federated Learning	94.83	93.47	93.83	91.67	90.12	90.50	93.74	92.38	92.77
Full AEF-IDS	96.74	95.83	96.02	93.92	92.34	92.75	95.87	94.52	94.82

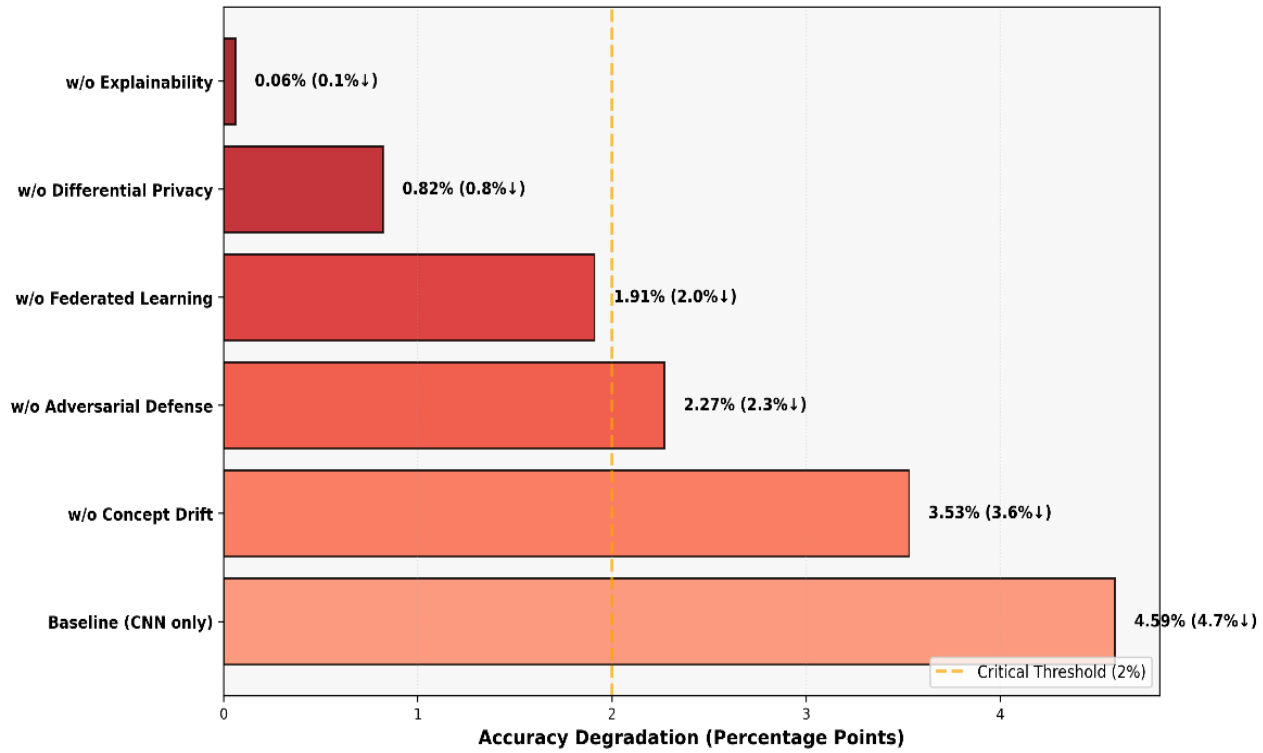

Figure 2. Performance Degradation by Component Removal

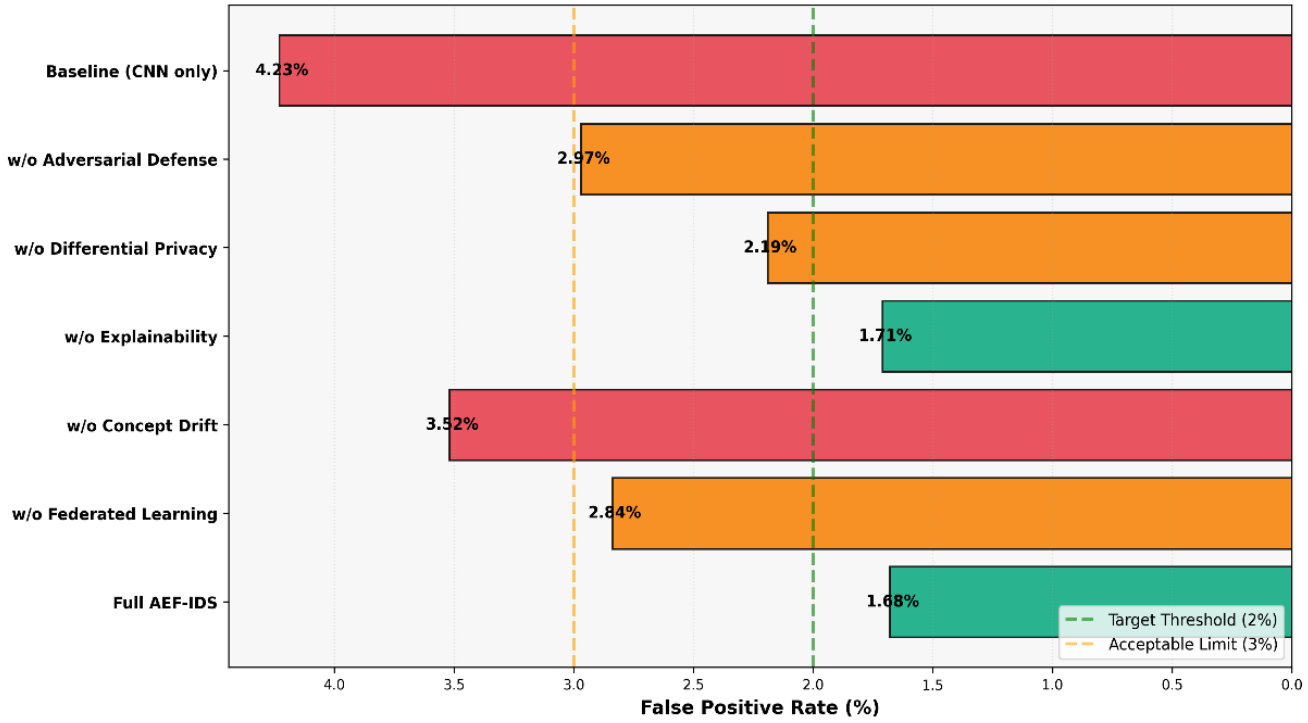

Figure 3. False Positive Rate Comparison Across Ablations (NSL-KDD dataset)

5.6 Explainability Analysis

The SHAP analysis (Figure 4) illustrates some uniform cross-dataset tendencies on feature significance. Connection-level attributes on NSL-KDD, including *srv_count* (0.77), *dst_host_srv_count* (0.76), and *dst_bytes* (0.76), dominate the attack attribution due to the service nature of the dataset traffic. In the case of UNSW-NB15, the high mean SHAP values are obtained by the features of the byte, such as *sbytes* (0.77) and *dbytes* (0.71), whereas CIC-IDS2018 is predominantly characterized by such statistics as *bwd_pkt_len_mean* (0.85) and *pkt_len_mean* (0.79). The LIME local explanation analysis (Figure 5) further corroborates these findings at the instance level, with *psh_flag_cnt* (0.694) and *syn_flag_cnt* (0.642) strongly supporting attack predictions in CIC-IDS2018, and *src_bytes* (0.550) and count (0.480) serving as primary positive contributors in NSL-KDD. The uniformity of SHAP global attribution and the LIME local explanation confirms the coherence of the interpretability framework and strengthens the belief in the model as a decision-making process.

6. Discussion

The experimental assessment shows that AEF-IDS consistently achieves greater superiority than traditional and deep learning intrusion detection techniques on all three standard datasets but has several limitations and practical issues that need to be analyzed carefully. While the proposed framework outperforms the baseline models such as CNN-IDS, Random Forest and LSTM-IDS in terms of classification process over the best baseline, the overall accuracy gains i.e., the difference remained relatively low ranging from 0.45% to 0.74% over the best baseline. This finding implies that the core of AEF-IDS is not just the incremental improvement of classification performance, but how these four components – federated learning, explainability, concept drift adaptation, and adversarial robustness – are brought together with a single framework. Recent research works have also shown that the new architectures of IDS should be capable of detecting intrusions under the constraints of being scalable, preserving privacy of the data being analyzed and transparent to the task at hand in a distributed computing setting, including the edge settings in a typical IoT deployment [4, 5, 14].

Both the concept drift adaptation mechanism and the other components of the proposed system are crucial for the proposed system and are also found to have several shortcomings under changing attack conditions in the experimental results. For the NSL-KDD, the system degrades accuracy by more than 7% after controlled distributional changes which occurs before adaptation. The latencies of the adaptation of the KS test-based drift detector for retraining are around 19 s, which can be too long for attacks that are moved to the zero-day domain or take easily adapted traffic patterns. These findings are echoed in recent research within the field of cybersecurity, emphasizing that in very dynamic networks, such as the IoT and 5G, the distribution of attacks may change continuously, making it difficult for statistical drift detection techniques to operate effectively [11, 26].

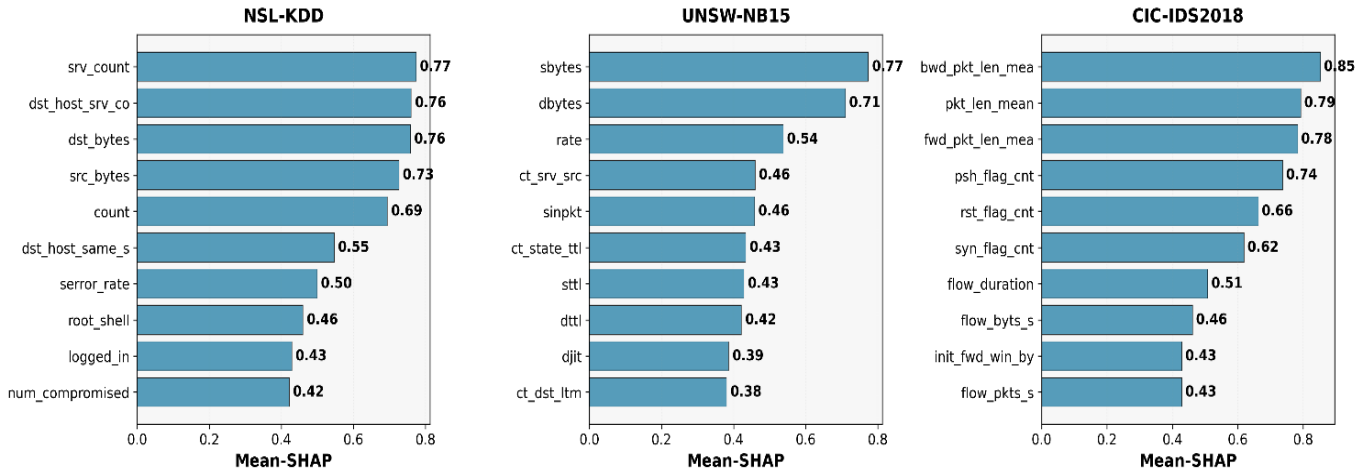

Figure 4. Cross-Dataset Feature Importance Comparison

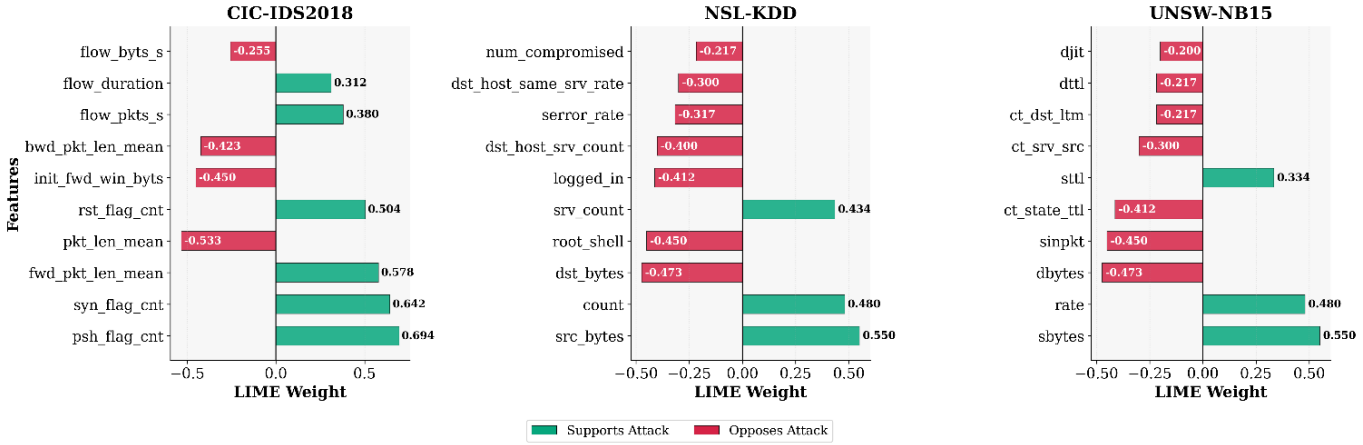

Figure 5. LIME Local Explanations Across Datasets (Attack Prediction)

Therefore, use of the adaptive retraining approach to enhance resilience has a downside in that it assumes statistical monitoring at regular intervals which would mean that in a large-scale operation, there may be a lack of responsive measures to address the situation in real time.

The adversarial robustness analysis also shows some significant restrictions of the framework. While AEF-IDS achieves relatively good attack resistance to FGSM, PGD, C&W and Deepfool perturbations, PGD attacks do affect the detection performance greatly across all datasets. The results show that the framework is vulnerable to higher order adversarial attacks, such as the adaptive and black-box attacks, since PGD is regarded as one of the best adverse attacks of order one. Adversarial machine learning is also a concern in the previous research of communication and cybersecurity systems, where the existing IDS methods may not be flexible to adapt to novel methods that are not included in the training set [29, 30]. Thus, although using adversarial training involves better robustness properties than those of traditional IDS models, further defenses for deployment towards mission critical infrastructures can be considered like certified robustness, ensemble adversarial learning, or transformer-based anomaly modeling.

Another restriction regards the configuration of federated learning adopted for the experiments. Standard FedAvg aggregation over just three participating clients is not representative of the statistical heterogeneity, communication instability, and Byzantine failure conditions that are generally found in real-world distributed settings. The diversity of clients and the distribution of data (IID and non-IID) have been shown to significantly affect convergence stability and fairness in recent federated IDS studies [9, 16]. Additionally, this is relatively low level of privacy, due to low privacy budget (ϵ = 10.0) chosen for the adopted differential privacy mechanism. Moreover, this is a relatively weak level of privacy, as compared to the higher degree of privacy, normally suggested for sensitive cybersecurity applications [17]. Thus, while a stronger Privacy Budget usually results in lesser utility of the model, future research on adaptive Privacy Preserving mechanisms that support the development of a tighter trade-off between privacy and detection is needed.

Overall, AEF-IDS is a significant step forward for the development of integrated, privacy-preserving and explainable intrusion detection for future networks. Despite that, the results showed that more enhancements are needed for the drift adaptation speed, adversarial generalization, federated scalability,

privacy, and formal guarantees before the framework could be deemed as suitable for the highly dynamic real-world cybersecurity domain [1, 2, 4, 29].

7. Conclusion

The Adaptive Explainable Federated Intrusion Detection System (AEF-IDS) suggested integrates privacy-preserving federated learning, KS-test-based concept drift detection, differential privacy, adversarial robustness, and multi-level explainability algorithms in a single graphical framework of next-generation network security. The experimental results on NSL-KDD, UNSW-NB15, and CIC-IDS2018 indicate that AEF-IDS has high detection accuracy (above 93%) and low false positive rates (below 3%), excellent AUC-ROC performance, and per-sample inference latency of no more than 50 ms, thus meeting both effectiveness and real-time requirements. The system has a high level of performance under strong white-box attacks, that is, FGSM, PGD, C&W, and DeepFool, and both SHAP and LIME present actionable, feature-level explanations of intrusions and benign traffic detected by the system. These features render AEF-IDS a viable option to be used in distributed and privacy-sensitive networks where raw information cannot be centralized. However, the research is restricted to using benchmark datasets, a limited number of adversarial threat models, and a fixed federated topology and hyperparameter scheme. Validations should be done in the future with large-scale real-world implementations, adaptive client interaction and communication timetable, wider adaptive and stealthy assault classes, and investigations of cross-topography overall across heterogeneous IoT, 5G, and industrial management infrastructures.

References

[1] C. Merlano, Enhancing cyber security through artificial intelligence and machine learning: a literature review. Journal of Cybersecurity, 6, (2024) 89. https://doi.org/10.32604/jcs.2024.056164

[2] I.H. Sarker, Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. Annals of Data Science, 10(6), (2023) 1473-1498. https://doi.org/10.1007/s40745-022-00444-2

[3] I.H. Sarker, CyberLearning: Effectiveness Analysis of Machine Learning Security Modeling to Detect Cyber-Anomalies and Multi-Attacks. Internet of Things, 14, (2021) 100393. https://doi.org/10.1016/j.iot.2021.100393

[4] A. Karunamurthy, K. Vijayan, P.R. Kshirsagar, K.T. Tan, An optimal federated learning-based intrusion detection for IoT environment. Scientific Reports, 15(1), (2025). https://doi.org/10.1038/s41598-025-93501-8

[5] H. Liao, M.Z. Murah, M.K. Hasan, A.H.M Aman, J. Fang, X. Hu, A.U.R. Khan, A survey of deep learning technologies for intrusion detection in internet of things. IEEE Access, 12, (2024) 4745-4761. https://doi.org/10.1109/ACCESS.2023.3349287

[6] L.A. Maghrabi, Automated network intrusion detection for internet of things: Security enhancements. IEEE Access, 12, (2024) 30839-30851. https://doi.org/10.1109/ACCESS.2024.3369237

[7] H.A.A. Hasan, M. Zolfy, Exploring lightweight deep learning techniques for intrusion detection systems in iot networks: A survey. Journal of Electrical Systems, 20(4s), (2024) 1944-1958. https://doi.org/10.52783/jes.2292

[8] A. Adamova, T. Zhukabayeva, N. Adamov, Machine learning algorithms for intrusion detection in IoT-enabled smart homes. Procedia Computer Science, 241, (2024) 427-432. https://doi.org/10.1016/j.procs.2024.08.059

[9] N. Albanbay, Y. Tursynbek, K. Graffi, R. Uskenbayeva, Z. Kalpeyeva, Z. Abilkaiyr, Y. Ayapov, Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study. Journal of Sensor and Actuator Networks, 14(4), (2025) 78. https://doi.org/10.3390/jsan14040078

[10] Y. Shewale, S. Kumar, S. Banait, Machine learning based intrusion detection in IoT network using MLP and LSTM. International Journal of Intelligent Systems and Applications in Engineering, 11(7S), (2023) 210-223. https://doi.org/10.17762/ijritcc.v11i2.6109

[11] Y. Guo, A review of machine learning-based zero-day attack detection: Challenges and future directions. Computer communications, 198, (2023) 175-185. https://doi.org/10.1016/j.comcom.2022.11.001

[12] H. Bangui, B. Buhnova, Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms. Computers and Electrical Engineering, 100, (2022) 107901. https://doi.org/10.1016/j.compeleceng.2022.107901

[13] S. Racherla, P. Sripathi, N. Faruqui, M.A, Kabir, M. Whaiduzzaman, S.A. Shah, Deep-IDS: a real-time intrusion detector for IoT nodes using deep learning. IEEE Access, 12, (2024) 63584-63597. https://doi.org/10.1109/ACCESS.2024.3396461

[14] A. AlHayan, J. Al-Muhtadi, Federated learning-powered real-time behavioral intrusion detection leveraging LSTM, attention, GANs, and large language models. Scientific Reports, 16, (2026). https://doi.org/10.1038/s41598-026-40763-5

[15] K. Ileri, Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods

optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks. International Journal of Machine Learning and Cybernetics, 16(9), (2025) 6937-6956. https://doi.org/10.1007/s13042-025-02654-5

[16] B. Yang, G. Zhang, K. Wang, A Federated Deep Transfer Learning Algorithm for Intrusion Detection. International Journal of Information Security and Privacy (IJISP), 19(1), (2025) 1-27. https://doi.org/10.4018/IJISP.387079

[17] A. Iričanin, O. Ristić, M. Milošević, (2024). Privacy-Preserving in Machine Learning: Differential Privacy Case Study. In 10th International Scientific Conference Technics, Informatics and Education-TIE 2024. Faculty of Technical Sciences Čačak, University of Kragujevac, 89-96. https://doi.org/10.46793/TIE24.089I

[18] S. Jain, V. Sharma, Decision Trees in Intrusion Detection: A Comparative Analysis of Machine Learning Techniques. International Journal of Telecommunication and Emerging Technologies, 11(1), (2025) 1-10.

[19] H. Rhachi, Y. Balboul, A. Bouayad, Enhanced anomaly detection in IoT networks using deep autoencoders with feature selection techniques. Sensors, 25(10), (2025) 3150. https://doi.org/10.3390/s25103150

[20] R. Golchha, A. Joshi, G.P. Gupta, Voting-based Ensemble Learning approach for Cyber Attacks Detection in Industrial Internet of Things. Procedia Computer Science, 218, (2023) 1752–1759. https://doi.org/10.1016/j.procs.2023.01.153

[21] S.M. Tseng, Y.Q. Wang, Y. C. Wang, Multi-class intrusion detection based on transformer for IoT networks using CIC-IoT-2023 dataset. Future Internet, 16(8), (2024) 284. https://doi.org/10.3390/fi16080284

[22] S. Subramani, M. Selvi, Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks. Optik, 273, (2023) 170419. https://doi.org/10.1016/j.ijleo.2022.170419

[23] L. Haitao, W. Ruimin, D.O.N.G Weiyu, J.I.A.N.G. Liehui, Semi-supervised Network Traffic Anomaly Detection Method Based on GRU. Computer Science, 50(03), (2023) 380-390.

[24] S.I. Popoola, Y. Tsado, A.A. Ogunjinmi, E. Sanchez-Velazquez, Y. Peng, D. B. Rawat, Multi-Stage Deep Learning for Intrusion Detection in Industrial Internet of Things. IEEE Access, 13, (2025) 60532 – 60555. https://doi.org/10.1109/ACCESS.2025.3557959

[25] R.M. Kawale, R.V. Patil, L.V. Patil, S.A. Mahajan, Performance evaluation of machine learning algorithms with fuzzy logic for intrusion detection in VANET network. Journal of Fuzzy Extension and Applications, 7(1), (2026) 312-331. https://doi.org/10.22105/jfea.2025.505777.1790

[26] A. Abdallah, A. Alkaabi, G. Alameri, S. H. Rafique, N. S. Musa, T. Murugan, Cloud Network Anomaly Detection Using Machine and Deep Learning Techniques - Recent Research Advancements. IEEE access, 12, (2024) 56749 – 56773. https://doi.org/10.1109/ACCESS.2024.3390844

[27] A.V. Potnurwar, V.K. Bongirwar, S. Ajani, N. Shelke, M. Dhone, N. Parati (Deep learning-based rule-based feature selection for intrusion detection in industrial Internet of Things networks. International Journal of Intelligent Systems and Applications in Engineering, 11(10s), 2023) 23-35.

[28] Y. Imrana, Y. Xiang, L. Ali, Z. Abdul-Rauf, A bidirectional LSTM deep learning approach for intrusion detection. Expert Systems with Applications, 185, (2021) 115524. https://doi.org/10.1016/j.eswa.2021.115524

[29] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, H.V. Poor, Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey. IEEE Communications Surveys & Tutorials, 25(4), (2023) 2245-2298. https://doi.org/10.1109/COMST.2023.3319492

[30] M. Al-Ajlan, M. Ykhlef, A Review of Generative Adversarial Networks for Intrusion Detection Systems: Advances, Challenges, and Future Directions. Computers, Materials & Continua, 81(2), (2024) 2053–2076. https://doi.org/10.32604/cmc.2024.055891

[31] L.S. Kumar, S.R. Nethi, R. Uyyala, P. Vurubindi, S.C. Narahari, A.K. Das, B.K. Vivekananda, M.J. Alenazi, Anomaly-based intrusion detection on benchmark datasets for network security: a comprehensive evaluation. Scientific Reports, 16(1), (2026) 8507. https://doi.org/10.1038/s41598-026-38317-w

[32] N. Farnaaz, M.A. Jabbar, Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, (2016) 213-217. https://doi.org/10.1016/j.procs.2016.06.047

[33] A. Gouveia, M. Correia, Network intrusion detection with XGBoost. Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS), Chapman and Hall/CRC.

[34] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, (2019) 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[35] J. Du, M. Xiao, Y. Li, S. Yu, NIDS-CNNLSTM:

Network intrusion detection classification model based on deep learning. IEEE Access, 11, (2023) 24808–24821. https://doi.org/10.1109/ACCESS.2023.3254915

[36] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy ICISSP, 1(2018), 108-116. https://doi.org/10.5220/0006639801080116

[37] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the KDD CUP 99 data set. IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, Canada. https://doi.org/10.1109/CISDA.2009.5356528

[38] N. Moustafa, J. Slay, (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Military communications and information systems conference (MilCIS), IEEE, Australia. https://doi.org/10.1109/MilCIS.2015.7348942

[39] A. Deshmukh, P.E. de la Rosa, R.V. Rodriguez, S. Dasari, Enhancing privacy in IoT-enabled digital infrastructure: Evaluating federated learning for intrusion and fraud detection. Sensors, 25(10), (2025) 3043. https://doi.org/10.3390/a18050294

Authors Contribution Statement

Rabins Porwal: Conceptualization, Methodology, Investigation, Writing – original draft. Manoj Singh Adhikari: Data curation, Software, Validation, Formal analysis. S. Keerthi: Supervision, Visualization, Writing – review & editing. Anil Kumar Yadav: Resources, Investigation, Formal analysis. Mahesh Babu Ketha: Validation, Formal analysis, Writing – review & editing. Piyush Verma: Investigation, Formal analysis, Visualization. Kunal: Project administration, Writing – review & editing, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes