Majestic International Journal of AI Innovations
ISSN: 3142-8428
Volume 1, 2026, **pp.** 1-16, **Article ID:** CM2601105003
DOI: 10.65080/mjai.v1.CM2601105003

CONTENT MAJESTIC PUBLISHER
OPEN ACCESS

Research Article

Federated Learning Framework for Privacy-Preserving Threat Detection in Distributed IT Systems

Imad Ullah[1, *], **Ibad Ullah**[1], **Naseer Ullah**[1]

[1]*Faculty of Computing, Riphah International University, Islamabad, Pakistan*

Article History

Received: 11 March, 2026 Revised: 01 April, 2026 Accepted: 11 May, 2026 Published: 26 May, 2026

Abstract:

Introduction: This study investigates the practicality and efficiency of federated learning as a privacy-preserving threat detection approach in distributed IT systems. The research is motivated by the limitations of centralized intrusion detection systems and increasing regulatory constraints on data sharing. The primary aim is to evaluate whether federated learning can achieve competitive detection performance while maintaining strict data privacy.

Methods: An end-to-end federated learning framework was implemented and evaluated using the UNSW-NB15 dataset, consisting of 257,673 network flow records with 36 traffic-related features. A deep neural network model was collaboratively trained across multiple non-IID clients using the Federated Averaging algorithm, ensuring that raw network traffic data remained local to each client. The federated model was compared with centralized deep learning, local-only training, and a centralized Random Forest classifier. In addition to predictive performance, system-level metrics such as communication efficiency, convergence behaviour, and deployment feasibility were analysed.

Results: Experimental results show that the federated model achieved competitive detection performance, closely approaching centralized deep learning while outperforming local-only training. The centralized Random Forest classifier achieved the highest predictive accuracy but required full data aggregation and therefore lacked privacy guarantees. The federated model demonstrated stable convergence within 20 communication rounds, with most performance gains achieved by round 15. Communication overhead remained modest at approximately 10.85 MB, and computational costs were manageable. Although federated training required more time than centralized deep learning, it preserved privacy by ensuring that raw network traffic remained within client environments.

Conclusion: The findings demonstrate that federated learning provides a practical and privacy-aware alternative to centralized intrusion detection systems. It effectively balances detection performance, communication efficiency, and reduced data exposure, making it suitable for deployment in distributed environments with realistic non-IID data conditions.

Keywords: Federated learning, intrusion detection, privacy-preserving machine learning, distributed IT systems, deep neural networks, UNSW-NB15, non-IID data, cybersecurity.

1. INTRODUCTION

Cloud The IT infrastructure has experienced a radical shift, which has become highly distributed due to cloud platforms, edge devices, and geographically distributed enterprise networks [1, 2]. Organizations are progressively using hybrid and multi-cloud systems, remote access, and interconnected digital ecosystems to facilitate scalability, stability, and operational effectiveness [3, 4]. Although this architectural redesign allows flexibility and performance, it also increases the area of the cyber-attack, and distributed IT systems become one of the main targets of more advanced threats.

In order to address these threats, intrusion detection and threat forecasting systems based on machine learning (ML) have become a key element of modern cybersecurity policies.

*Address correspondence to this author at Faculty of Computing, Riphah International University, Islamabad, Pakistan; E-mail: Imaduom@gmail.com

Cite as: Ullah I, Ullah I, Ullah N. Federated Learning Framework for Privacy-Preserving Threat Detection in Distributed IT Systems. *Majestic International Journal of AI Innovations*, 2026; *1*, 1–16. Article ID: CM2601105003. https://doi.org/10.65080/mjai.v1.CM2601105003

By learning patterns from historical network traffic and system logs, ML models detect abnormal and malicious behaviour more accurately than rule-based systems [5, 6]. Most of the current ML-based systems of intrusion detection (IDS) are, however, based on centralized learning models whereby raw security data is gathered into one repository.

The centralized collection of data is now being limited more by privacy laws, corporate policies, and trust limits [7]. There are also laws like the General Data Protection Regulation (GDPR) and industry-related compliance laws that limit the distribution of sensitive network logs, user behavioural traces, and operational metadata [8]. Even in the absence of regulatory barriers, organizations are often reluctant to disclose raw security data due to intellectual property concerns, reputational risk, and potential misuse. Because of this, security data is often isolated by departments, subsidiaries, or partner organizations, which reduces the usefulness of centralized analytics.

These limitations have inspired an increasing amount of interest in privacy-preserving learning paradigms, especially federated learning (FL). In federated learning, several parties jointly train a global model, but retain raw data on a local scale at each location. Participants, instead of sharing data, share model parameters or updates, which diminishes the chances of directly exposing data. This paradigm is particularly attractive to distributed IT systems, where the cooperation across the boundaries between organizations can enhance threat awareness without breaking privacy or governance restrictions. However, the federated learning implementation for real-world and practical threat detection in distributed settings is still a challenge that is open and poorly investigated.

According to [9], machine learning has proved to be effective in cybersecurity, the existing intrusion detection systems have inherent challenges when used in distributed IT systems. The centralized IDS designs implicitly presuppose the possibility of collecting, storing, and processing security data from all sources in one location. This is becoming an unrealistic assumption in the context of regulatory constraints, decentralized ownership and heterogeneous infrastructure.

Some serious threats are posed by centralized learning strategies. To start with, raw traffic information and logs can be directly aggregated, which can breach the privacy rules or the internal security policies [10]. Second, repositories that are centralized are valuable assets, and the damage of data breaches becomes even greater [11]. Third, centralized systems demand high-bandwidth data transfer and continuous synchronisation, which can be impractical or expensive in large-scale distributed deployment [12].

In the other extreme, local-only learning, in which each organization or site trains its own isolated model, also has its fair share of downsides. The local models have the disadvantage of low visibility, since they are trained on small and possibly biased data sets [13]. Consequently, they tend to miss new or unusual attack patterns, which can only be discovered when the information is consolidated across a number of sites. Such fragmentation results in inconsistency in the performance of threat detection and compromises the system-wide security position.

To add to these issues, security data in organizations is non-independent and identically distributed (non-IID). Various locations have dissimilar traffic patterns, workloads, and attack profiles, as determined by their roles, configurations and exposure to threats. This heterogeneity makes collaborative learning difficult, and model convergence and performance are, in most cases, poor when not specifically taken care of. This creates a need for learning frameworks that operate effectively under non-IID conditions while respecting privacy and trust constraints in distributed IT systems.

There is a lot of literature on machine learning-based intrusion detection, but to a large extent, it is accuracy-focused and presupposes centralized data access. Most of the suggested IDS models are only tested using predictive performance measures without much thought of implementation, privacy or cost of the system. Moreover, much of the previous literature is based on excessively simplistic data that is not representative of modern network traffic.

More current research has investigated federated learning for use in cybersecurity, such as intrusion detection. Although these attempts prove the theoretical viability of FL-based IDS, there are still a number of limitations. A lot of FL-IDS research is theoretical or based on experimental systems on a small scale. Assessments do not take into account realistic non-IID data distributions, communication overhead, and convergence behaviour, which are factors that strongly affect practical implementation. Moreover, few studies have proven federated IDS solutions on recent and extensive data like UNSW-NB15, which reveals various types of attacks and the actual traffic features. Consequently, no end-to-end empirically validated federated learning architectures that collectively consider the accuracy of threat detection, privacy protection, system-level effectiveness, and data heterogeneity on distributed IT settings have been proposed. Such a gap spurs the current research.

To fill the given gaps, the following key contributions were made in this paper:

1. System-level federated IDS evaluation, analysing convergence, communication overhead, and round efficiency.
2. Empirical analysis of non-IID impact, demonstrating stable convergence under heterogeneous client distributions.
3. Communication-efficient federated IDS evaluation, explicitly quantifying communication overhead and convergence rounds under realistic non-IID client distributions.
4. Reproducible federated intrusion detection pipeline using UNSW-NB15 under realistic deployment constraints.

These contributions collectively distinguish this work from prior FL-IDS studies, which typically focus on detection accuracy alone without jointly addressing system-level efficiency, non-IID convergence behaviour, communication overhead, and deployment feasibility within a single reproducible evaluation. While FedAvg and DNN architectures are individually well-established, their integration within an end-to-end empirically validated federated pipeline under

realistic heterogeneous conditions with explicit quantification of communication and computation cost represents the primary novelty of this study.

The paper mainly explores the possibility of using federated learning to facilitate an effective and privacy-conscious threat detection in distributed IT systems. In particular, the research questions that the study attempted to answer included:

RQ1: Can federated deep neural networks achieve competitive threat detection performance compared to centralized learning approaches?

RQ2: How does non-IID data distribution across distributed clients affect federated learning convergence and accuracy?

RQ3: What is the system-level costs, in terms of communication and computation, associated with federated intrusion detection?

RQ4: What privacy–utility trade-offs arise when applying federated learning to threat detection tasks?

2. LITERATURE REVIEW

2.1. Machine Learning and Deep Learning for Intrusion Detection

The IDS has largely developed out of signature-based and rule-driven systems to data-driven systems that can recognize sophisticated and hitherto unseen attack patterns [14]. Conventional IDS methods, although useful in the background of known threats, are incapable of adapting to the dynamic threat environment and the growing size of contemporary network traffic [15, 16]. This shortcoming has induced the use of machine learning (ML) methods, which acquire discriminative patterns directly from data and have the ability to extrapolate beyond prescribed rules.

The initial ML-based IDS solutions were based on classical algorithms, including decision trees, support vector machines, and ensembles [15]. The techniques proved to be more accurate in detection than rule-based systems, especially in the detection of anomalies. Nevertheless, they are typically limited in their ability to represent complex and high-dimensional relationships in network traffic data due to the need to feature engineer and capacity limitations. These constraints were emphasized as the network environments became more heterogeneous and dynamic in terms of traffic patterns.

The most recent developments in deep learning have also changed the field of IDS research. Convolutional architectures, deep neural networks (DNNs) and recurrent models have been demonstrated to learn non-linear dependencies and time-dependent correlations of network flows [17]. DNN-based IDS models are especially suitable for tabular network traffic datasets, where interactions between multiple features are the drivers of attack behaviour, as opposed to individual indicators [18, 19]. Empirical studies consistently show that deep learning outperforms traditional machine learning methods.

Although these have been made, most deep learning-based IDS studies have centralized assumptions regarding access to training data. The models are usually trained on aggregated traffic logs, which are obtained through several sources, and an assumption that is becoming unrealistic in distributed IT systems [19, 20]. As a result, deep learning helps to increase the detection power, but it does not help to resolve the underlying issues of privacy, trust, and data governance that occur in practice.

2.2. Centralized Learning Limitations in Distributed IT Environments

The centralized learning architectures are still prevalent in the IDS literature because of their conceptual ease and excellent empirical results. Centralized models have the advantage of having an all-encompassing view of various attack patterns by consolidating the data from a variety of sources [21]. Nonetheless, there are major practical and security-related issues presented by this paradigm when it is used in relation to a distributed IT system [22].

First, the principle of data aggregation in a centralized manner is in direct contradiction with the privacy laws and the internal governance policies [10]. Network traffic logs can consist of sensitive user, application or organizational process information. Moving such data to a central point enhances the chances of data leakage and non-compliance with regulations. Second, centralized repositories generate desirable targets to the adversaries, increasing the magnitude of the consequences of a successful intrusion [11]. Third, the ongoing exchange of high-volume network traffic data is very expensive in terms of bandwidth and infrastructure expenses [12].

Centralized approaches also conceal the heterogeneity of distributed environments as seen in a modelling view. Non-IID characteristics in security data are frequently observed when data is collected by different organizations or network segments because of the difference in the workloads, configurations and vulnerability to threats. Aggregation [23]. This can cause the loss of local site-specific patterns, which would diminish the applications of local threat detection. The limitations highlight the necessity of alternative learning paradigms, which would be effective to work without central data gathering.

2.3. Federated Learning: Principles and Security Applications

Federated learning has become a potential paradigm of collaborative model training under privacy conditions. In FL, several clients use local models to train local data and periodically provide updates on those models to a central server, which combines the updates to create a global model [24]. Raw data never leave the client environment, and sensitive information is not directly exposed.

Federated learning was initially suggested in mobile and edge computing situations, but it has since been researched in many other areas, such as healthcare, finance, and cybersecurity. FL provides a promising mechanism for sharing threat intelligence across organizational boundaries and maintaining the locality of data in the context of security analytics. FL allows collective learning without the need to explicitly share data by aggregating learned representations instead of raw logs.

Nevertheless, federated learning comes with its own set of issues. The non-IID data distributions among clients may slow the convergence and deteriorate the performance of the model [25]. It can add a lot of overhead to communication between clients and the server, especially with deep learning models. Also, FL decreases the direct exposure of the data; however, the information leakage occurs when the model updates itself [26]. These issues require stringent empirical testing in the use of FL to security sensitive applications like intrusion detection.

2.4. Federated Learning for Intrusion Detection Systems

An emerging body of work has investigated the application of federated learning to IDS. The current literature on FL-IDS typically shows that federated models can be superior to local-only models and can achieve the performance of centralized systems in some circumstances [27]. These results confirm the effectiveness of FL as a privacy-saving substitute for cooperative intrusion detection.

However, there are still a number of weaknesses in the existing literature. Numerous studies test FL-IDS techniques in simplified experimental conditions, e.g., balanced data partitions or a few simulated clients [28, 29]. Practically, security data in organizations is very heterogeneous and realistic non-IID conditions are not often discussed in detail [29]. Besides, performance appraisal commonly pays too much attention to accuracy-based measures, but very little consideration to convergence behaviour, communication cost, or computational overhead.

A second significant limitation is the selection of the dataset. Much of the FL-IDS literature is based on old standards that are no longer relevant to modern attack vectors and network actions. The applicability to real-world conditions is not as high, since modern datasets, with different types of attack and realistic traffic properties, are used rather sporadically [30].

2.5. UNSW-NB15 Dataset in IDS Research

The UNSW-NB15 dataset is an important improvement of the previous intrusion detection benchmarks. It captures contemporary network traffic created under controlled conditions, which are realistic, but it contains a wide range of categories of attack, as well as normal behaviour [31]. The dataset is rich in features (including flow-based, protocol-level, and content-related) and is therefore appropriate to test the performance of sophisticated ML and deep learning models.

Previous IDS research on UNSW-NB15 has shown that it is viable in training and assessing the intrusion detection models [32]. Nevertheless, the majority of the available literature uses centralized learning paradigms and thus shares the privacy and scalability drawbacks mentioned above. The existing body of work that has covered federated learning with UNSW-NB15 is very limited, with those studies that do not typically analyse the system-level or assume realistic heterogeneous clients.

This is a major gap, especially considering that the dataset is suitable for modelling the distributed environment where various clients can be experiencing varying subsets of the traffic and types of attacks. The application of UNSW-NB15 in a federated environment offers a chance to test the performance of IDS in an environment that is close to real-life distributed IT systems.

2.6. Research Positioning

Although machine learning and deep learning have significantly advanced the intrusion detection processes, the current IDS research is full of centralized assumptions that cannot be used in contemporary distributed IT settings. Federated learning provides a more promising solution since it allows model training together without the exchange of raw data. Nevertheless, the existing FL-IDS literature is constrained by simplified assumptions, old-fashioned datasets, and a small number of evaluation criteria.

This paper addresses these weaknesses by introducing an end-to-end federated learning system to detect malicious traffic on the UNSW-NB15 dataset in realistic non-IID scenarios. With the addition of system-level metrics, privacy considerations, and predictive performance, the proposed solution advances the state of the art in practical and privacy-aware intrusion detection within distributed IT systems. Several important related research threads have emerged in the recent literature that contextualise and motivate the design choices made in this work.

From a privacy-preservation perspective, several recent studies have explored stronger formal mechanisms within FL-IDS frameworks. Differential privacy (DP) has been integrated into federated intrusion detection pipelines through calibrated noise injection and privacy budget accounting, providing measurable guarantees against gradient-based inference attacks [27, 29]. Secure aggregation protocols have been applied to prevent the central server from observing individual client updates, offering an additional layer of confidentiality. In contrast, the present study deliberately omits these mechanisms to serve as a controlled baseline that isolates the effects of federated averaging and non-IID data distribution on system performance, without the confounding influence of privacy noise on model convergence.

Regarding non-IID heterogeneity, recent work has proposed alternative aggregation strategies that improve convergence under heterogeneous client data. FedProx introduces a proximal regularisation term that penalises local models for deviating too far from the global model, improving stability under heterogeneity. SCAFFOLD uses variance reduction through control variates to correct for client drift caused by non-IID distributions. These methods demonstrate measurable improvements over standard FedAvg in heterogeneous settings. The present study adopts FedAvg as a deliberate baseline choice to enable clean and interpretable system-level analysis; future work should benchmark the proposed pipeline against FedProx and SCAFFOLD to quantify any performance and convergence gains under the non-IID conditions used here.

3. METHODOLOGY

3.1. Framework Overview

The proposed methodology takes the federated learning paradigm to allow privacy-preserving threat detection over

distributed IT systems. The objective is to jointly train a global intrusion detection model without exchanging raw network traffic among participants. Local model training by each distributed client, which is a separate organizational node or network segment, is done on its own data, and a central coordinating server combines the learned parameters to update a shared global model (Fig. **1**).

Let K denote the total number of participating clients. Each client $k \in \{1,2,\dots,K\}$ possesses a local dataset $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, where $x_i^k \in \mathbb{R}^d$ represents the feature vector of a network flow and $y_i^k \in \{0,1\}$ denotes the corresponding class label indicating normal or malicious activity. The total dataset size is given by $N = \sum_{k=1}^{K} n_k$. Importantly, the local datasets are assumed to be non-independent and non-identically distributed, reflecting realistic heterogeneity across distributed IT environments.

The iteration of the learning process is through a series of federated communication rounds. The global model parameters are sent to the clients in every round, and they are locally updated with local data and aggregated at the server to create a better global model. The process goes on until convergence or any predetermined number of rounds is achieved.

3.2. Dataset Preparation and Feature Representation

This paper uses the UNSW-NB15 dataset as the empirical evidence. It includes network flow logs simulated in a controlled but realistic environment, both benign traffic and a variety of cyber-attacks. The flows are characterized by a set of numerical and categorical characteristics that represent the statistics of packets, numbers of bytes, timing characteristics, and protocol-level characteristics.

One-hot encoding converts categorical features into numerical values to ensure that they can be used in models based on the neural network. Minimax scaling is used to normalize numerical features so as to have stable gradient-based optimization [33]. Let x_{ij} denote the value of the feature j for the sample i. Feature normalisation is defined as

$$\tilde{x}_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{1}$$

where $\min(x_j)$ and $\max(x_j)$ denote the minimum and maximum values of the feature j across the training data. This transformation reduces feature-scale disparities and improves training convergence.

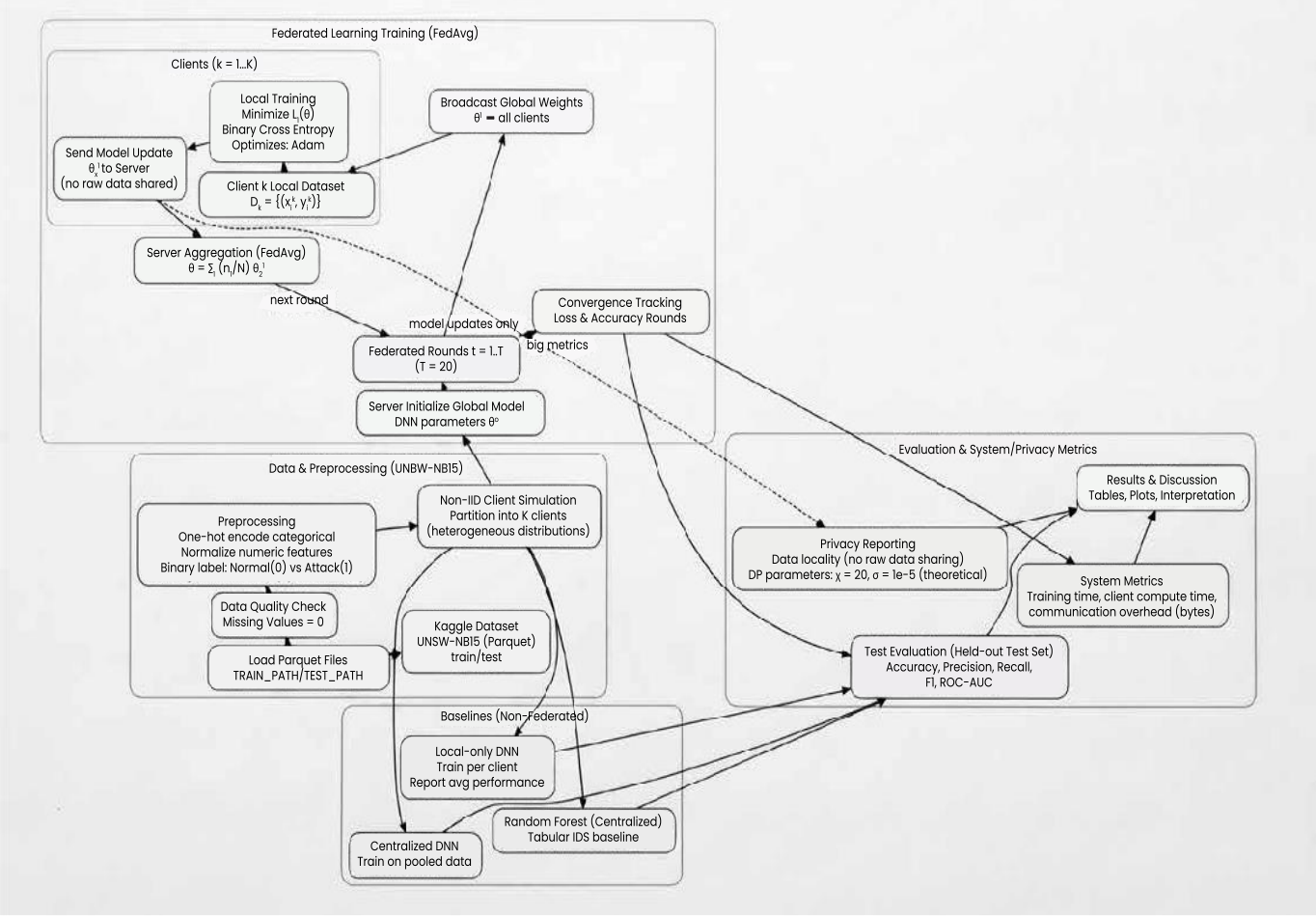

Fig. (1). Proposed framework.

The original multi-class attack labels are converted to a binary classification problem, in which normal traffic is assigned label 0 and malicious traffic is assigned label 1. This expression is in line with the threat detection aim of separating benign and malicious activity in distributed systems.

3.3. Non-IID Distributed Client Simulation

In order to achieve realistic distributed IT systems, the pre-processed dataset is divided in Kclient-specific subsets that have heterogeneous statistical properties. In contrast to random or uniform splitting, the partitioning strategy is intended to bring non-IID distributions to clients. Every client had different shares of attack and normal traffic, and different exposure to particular traffic features.

Formally, the data distribution at the client k is denoted by $P_k(x, y)$, which differs from the global distribution $P(x, y)$. That is,

$$P_k(x, y) \neq P_{k'}(x, y), \forall k \neq k'. \quad (2)$$

This heterogeneity reflects real-world conditions in which different organizations or network segments experience unique workloads and threat profiles. This is essential to assess federated learning in these conditions because non-IID data is known to adversely impact model convergence and stability.

3.4. Threat Detection Model Architecture

The base learner used in this paper is a fully connected deep neural network that is adapted to tabular network traffic data. Let $f(x; \theta)$ denote the neural network parameterized by θ. The network is composed of the input layer that is equal to the dimension of features, a series of hidden layers that have the rectified linear unit activations, and one output neuron that has the sigmoid activation to provide a probability estimate of the malicious activity.

Given an input feature vector x, the network output is defined as

$$\hat{y} = \sigma\left(W^{(L)} h^{(L-1)} + b^{(L)}\right), \quad (3)$$

where $h^{(l)} = \phi\left(W^{(l)} h^{(l-1)} + b^{(l)}\right)$ for $l = 1, \ldots, L-1$, $\phi(\cdot)$ denotes the ReLU activation function, $\sigma(\cdot)$ denotes the sigmoid function, and L is the total number of layers. Dropout regularization is applied to hidden layers to reduce overfitting and enhance generalisation.

The model is trained using the binary cross-entropy loss, which is widely adopted for supervised binary intrusion detection tasks [34].

$$\mathcal{L}(\theta) = -\frac{1}{n_k} \sum_{i=1}^{n_k} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \quad (4)$$

where $\hat{y}_i = f(x_i; \theta)$ is the predicted probability for the sample i.

3.5. Federated Learning Optimization Procedure

Federated learning is conducted using the Federated Averaging (FedAvg) algorithm, which is the canonical optimisation strategy for distributed non-IID federated settings [35]. FedAvg was deliberately selected to enable controlled analysis of communication overhead and convergence behaviour without introducing additional algorithmic complexity from alternative aggregation methods. At the beginning of the communication round t, the server holds the global model parameters $\theta^{(t)}$ and distributes them to all participating clients. Each client initialises its local model with $\theta^{(t)}$ and performs local optimization using its private dataset.

Local training for the client k means the reduction of the empirical risk

$$\mathcal{L}_k(\theta) = \frac{1}{n_k} \sum_{(x_i^k, y_i^k) \in \mathcal{D}_k} \ell\left(f\left(x_i^k; \theta\right), y_i^k\right), \quad (5)$$

where $\ell(\cdot)$ denotes the binary cross-entropy loss. Gradient-based optimization using the Adam optimizer is applied for a fixed number of local epochs, yielding updated parameters $\theta_k^{(t+1)}$.

Each client also sends its model updates to the server upon completion of local training. The server combines the updates to calculate the subsequent global model based on

$$\theta^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} \theta_k^{(t+1)}. \quad (6)$$

This weighted averaging scheme ensures that clients with larger local datasets contribute proportionally more to the global model update, reflecting their empirical risk. Under non-IID data distributions, the weighted aggregation in FedAvg mitigates local bias by proportionally incorporating client-specific empirical risks, although convergence may still be slower compared to IID settings.

3.6. Baseline Learning Approaches

Three baseline approaches are used to contextualize the performance of the proposed federated framework. The union of all client datasets is used to train a centralized deep neural network, which corresponds to an upper-bound performance case where privacy is not a concern. Another method is also considered, which is a local-only method where each client learns an independent DNN using its own data, and performance metrics are averaged across clients. Furthermore, a Random Forest classifier is trained in a centralized fashion to offer a solid non-neural baseline of tabular intrusion detection. The comparison of these methods allows evaluating in detail trade-offs between system-level efficiency, predictive performance, and privacy preservation.

3.7. Privacy-Preserving Considerations

This work adopts a privacy-aware federated learning approach based on strict data locality. No formal differential

privacy guarantees are provided, as no noise injection, gradient clipping, or privacy accounting mechanisms are implemented. Consequently, the framework prioritises regulatory compliance and reduced data exposure rather than formal differential privacy. Accordingly, any references to privacy in this study refer strictly to data locality and minimized data exposure, not to formal differential privacy guarantees or privacy budget accounting.

3.8. Evaluation Protocol

The performance of the models is measured on a held-out test set with common classification measures, such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve. Besides predictive performance, system-level metrics include federated training time, communication overhead and client-side computation cost, are logged. Such a multi-dimensional evaluation plan allowed the suggested structure to be assessed not only based on detection capacity but also regarding scalability, efficiency, and feasibility for implementation in distributed IT systems.

3.9. Implementation

3.9.1. Experimental Environment and Software Stack

The proposed federated learning system was developed on top of the Python programming language and run in a Google Colab environment to make it accessible and reproducible. Google Colab is an implementation platform based on the cloud with customizable computational power, which makes it appropriate to simulate distributed learning processes without local dedicated infrastructure. Python 3.x was used in all the experiments, and the base data processing and numerical computation were done by means of the NumPy and Pandas libraries. TensorFlow and Keras were used to implement model development and training and offer efficient abstractions to build and optimize deep neural networks. Other functions of the Scikit-learn library were used for preprocessing, baseline model, and evaluation metrics.

UNSW-NB15 data was saved in Apache Parquet format to allow loading it efficiently and using it in memory. Training and testing sets were separate files, and it was clear where the model development and evaluation took place. The whole experimental flow was run in a single Colab notebook, which enabled the end-to-end reproduction of the results between loading the data and generating the results.

3.9.2. Data Loading and Preprocessing Pipeline

The training and testing datasets were loaded into memory when initialized using the Parquet forms of the datasets. The data format is both numerical and categorical, and, therefore, it undergoes thorough pre-processing before training a model. Categorical variables like protocol type, service, and connection state were converted into numerical values through one-hot encoding so that the variables could be compatible with the neural network structure.

Minimum maximum scaling, as outlined in the methodology section, normalized the numerical features in order to reduce scale differences between the features, e.g., the number of packets, volume of bytes, and flow rates. This normalization was necessary to make the process of optimization based on gradients stable and helped to ensure uniform convergence behaviour between federation rounds.

The categories of attack were coded into a binary classification objective where normal traffic and malicious activity were discriminated. This expression can be reconciled with the main aim of threat detection in distributed IT systems, where early identification of anomalous behaviour is frequently valued over more detailed attack identification.

3.9.3. Federated Training Workflow

The paradigm of implementing the federated learning process consisted of a synchronous, round-based training paradigm. The global model parameters were announced to all the participating clients at the start of every federated round. All the clients represented a local instance of the deep neural network and started it with the received global parameters.

Local training was done on its own and dataset partition on each client. A fixed number of local epochs were trained with the Adam optimizer, and the gradient updates were calculated in batches to minimize the binary cross-entropy loss. The local training setup was maintained for clients to maintain consistent updates and stable aggregation.

Once local training was completed, every client sent its refined model parameters to the central server. There was no exchange of raw data samples, feature values or labels at any given point in this process. The updates received by the server were aggregated by weighted federated averaging, with each client contributing proportionally to the size of its local dataset. The updated global model was then performed with the aggregated parameters, and this completed one round of federated communication.

This process was repeated twenty times, which was the number of federated rounds chosen based on a preliminary convergence test. After every round, the progress indicators were recorded to track the completion of training and proper synchronization of the clients and the server.

3.9.4. Baseline Model Implementation

Three baseline learning strategies were applied with an identical pre-processing pipeline and assessment protocol in order to facilitate meaningful comparison. The joint training data of all clients was used to train a centralized deep neural network, which is an upper-bound case where privacy is no longer an issue. The same architecture and optimization settings used in the federated DNN were used in this model to provide a fair comparison.

Furthermore, a local-only learning setup was also tested, where individual clients trained their own independent deep neural network with local data only. This resulted in an average of performance measures per client to give an estimate of isolated learning performance in distributed settings.

Moreover, centralized data was used to train a Random Forest classifier that was a powerful non-neural intrusion detection benchmark in tabular data. The Random Forest model

was set up to have quite a number of trees to represent the interactions of the features, and still have a reasonable amount of time to train it.

3.9.5. Evaluation and Metric Computation

The test data was evaluated using a model to provide an unbiased evaluation of performance. Measures of standard classification, such as accuracy, precision, recall, F1-score and the area under the receiver operating characteristic curve, were calculated. All these measures would reflect on the detection effectiveness and sensitivity to class imbalance, which are paramount factors in intrusion detection tasks.

Besides predictive performance, system-level measures were also taken during the process of training. Federated training time was recorded as the total time it took to go through all communication rounds, and client-side computation time was recorded to determine the per-round training cost. It was estimated that communication overhead is the sum of the volume of model parameters exchanged between the clients and the server during all the rounds.

No differential privacy budget was computed, as no differential privacy mechanism (e.g., noise injection, gradient clipping, or privacy accounting) was implemented in the proposed framework. Although the current implementation did not use formal noise injection, the parameters that were reported can be used as a baseline for the privacy-utility trade-offs of the proposed framework.

3.9.6. Reproducibility and Practical Deployment Considerations

One of the design objectives of the implementation was reproducibility. The Colab notebook contained all the preprocessing, model configurations, and training procedures and could be completely repeated to achieve independent results. Random seeds were manipulated where possible to minimize run-to-run variability. In terms of deployment, the application shows that federated intrusion detection based on deep neural networks can be computationally efficient with relatively small resource demands. Training times were low on the client side, and there was low overhead on communication because of just exchanging model parameters as opposed to large volumes of traffic data. These features imply that the suggested framework can be scaled to real-world distributed IT systems, such as the settings where bandwidth is limited, or the capabilities of the computation are minimal.

4. RESULTS AND DISCUSSION

4.1. Exploratory Findings

The exploratory analysis of the UNSW-NB15 dataset demonstrates that there are various structural properties that have a direct impact on the performance and stability of intrusion detection models. According to descriptive statistics provided in Table **1**, a large number of traffic-related characteristics have the characteristics of heavy-tailed distributions with significant variances, especially when it comes to counts of packets, bytes, and transmission rates. Attributes like source bytes, destination bytes and traffic rate can vary across many orders of magnitude, which is indicative of bursty and heterogeneous real network traffic [31, 32].

This is visually depicted in Fig. (**2**), demonstrating a skewed distribution of flow durations to the right and (Fig. **3**), demonstrating a log-scaled boxplot of the volumes of source bytes, indicating the extreme outliers. These kinds of traffic patterns are typical of operational enterprise environments, where normal traffic is intersected with bursts of high volume, usually of scanning, probing, or denial-of-service activity [14, 15, 31]. The strong skewness and the extreme values in the data support the need to normalize features and encourage the application of deep neural networks with the ability to learn strong nonlinear decision boundaries [15, 16, 18].

The distribution of the labels also points to the unequal dataset. According to Table **2** and (Fig. **4**), the malicious traffic is a larger percentage of records as compared to normal traffic. This skew provides difficulties to classification models, especially in terms of recall and false-negative rates, which are important measures in intrusion detection [15, 16]. This imbalance is further intensified in a federated learning viewpoint when data is divided among clients, since each client might be seeing significantly different proportions of classes and traffic patterns [23, 25].

Table 1. Descriptive statistics of key numerical features.

Feature	Mean	Std. Dev	Min	25%	Median	75%	Max
dur	1.2500	5.9700	0.0000	0.0000	0.0040	0.6860	59.9900
spkts	19.7800	135.9500	1.0000	2.0000	4.0000	12.0000	10646.0000
dpkts	18.5100	111.9900	0.0000	0.0000	2.0000	10.0000	11018.0000
sbytes	8572.9500	173773.9000	24.0000	114.0000	528.0000	1362.0000	14355770.0000
dbytes	14387.2900	146199.3000	0.0000	0.0000	178.0000	1064.0000	14657530.0000
rate	91253.9000	160376.9000	0.0000	30.8000	2955.7000	125000.0000	1000000.0000
sload	7.0600×10^{7}	1.8600×10^{8}	0.0000	1.2300×10^{4}	7.4400×10^{5}	8.0000×10^{7}	5.9900×10^{9}
dload	6.5800×10^{5}	2.4100×10^{6}	0.0000	0.0000	1747.0000	22105.0000	2.2400×10^{7}

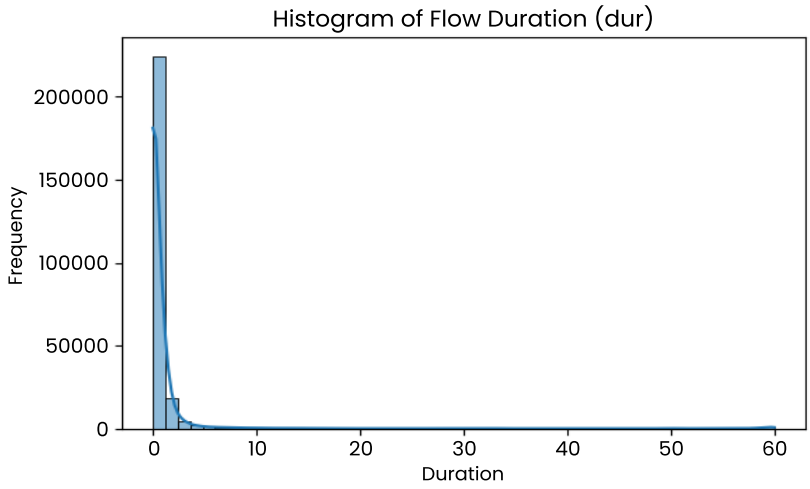

Fig. (2). Histogram of flow duration.

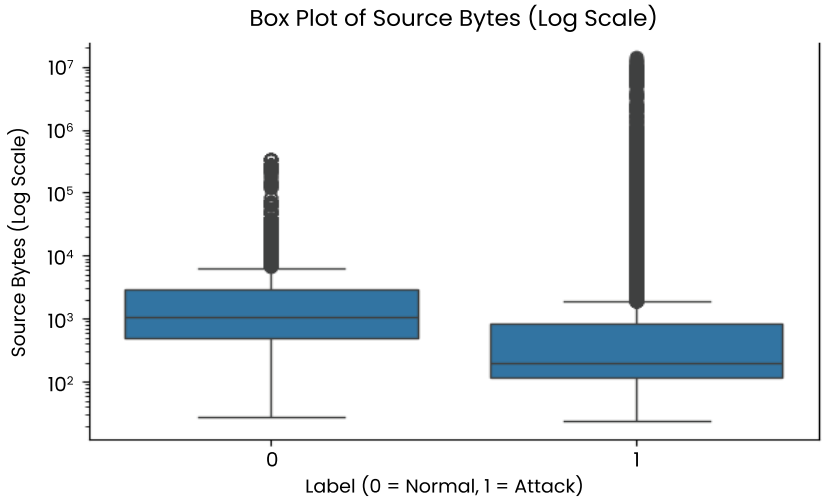

Fig. (3). Box plot of source bytes (log scale).

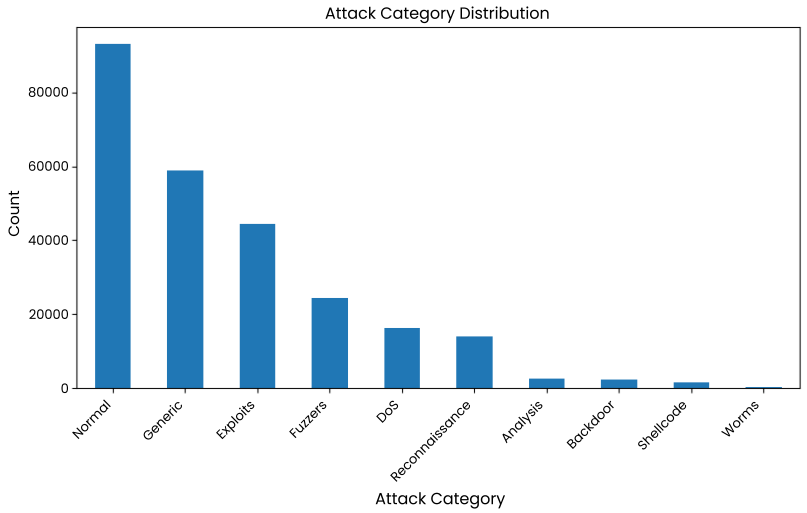

Fig. (4). Attack category distribution.

Table 2. Label distribution.

Class	Count	Percentage
Normal (0)	~92,000	~35.8%
Attack (1)	~165,673	~64.2%

The correlation analysis can give more insight into the interaction of features. As indicated by the correlation heatmap presented in Fig. (**5**), there is a strong dependence between a variety of protocol-level and transport-level features, and there is no feature that shows a dominant influence on the label of the attack. Rather, the behaviour of attacks is expressed in the form of multifaceted interactions between several attributes, which proves the appropriateness of deep learning solutions over linear or rule-based ones [15, 16, 18]. These exploratory results support the modelling and evaluation decisions adopted in this study.

4.2. Federated Learning Convergence Behaviour

The convergence behaviour of the federated deep neural network gives important information on whether collaborative learning is possible given non-IID conditions of data. The training loss is monotonic and stable in twenty federated communication rounds, as presented in Fig. (**6**), which suggests that locally learned representations are effectively integrated into a consistent global model. It is especially interesting to note that there are no oscillatory or divergent loss patterns, which non-IID data distributions are often attributed to being unstable under federated optimization [19, 25, 29].

Model accuracy shows the same trend, and it increases with each successive round until it reaches the end of the training period, as shown in Fig. (**7**). The convergence summaries are presented in Table **3**, which indicates that most of the performance gains are obtained during the first fifteen rounds. Further than this position, marginal gains are negligible, which implies that the federated model quickly learns the global patterns of global threat, despite the heterogeneity of clients. System design-wise, this outcome has the implication that communication rounds can be bounded without significantly affecting performance, which is significant with regard to implementation in bandwidth-constrained settings.

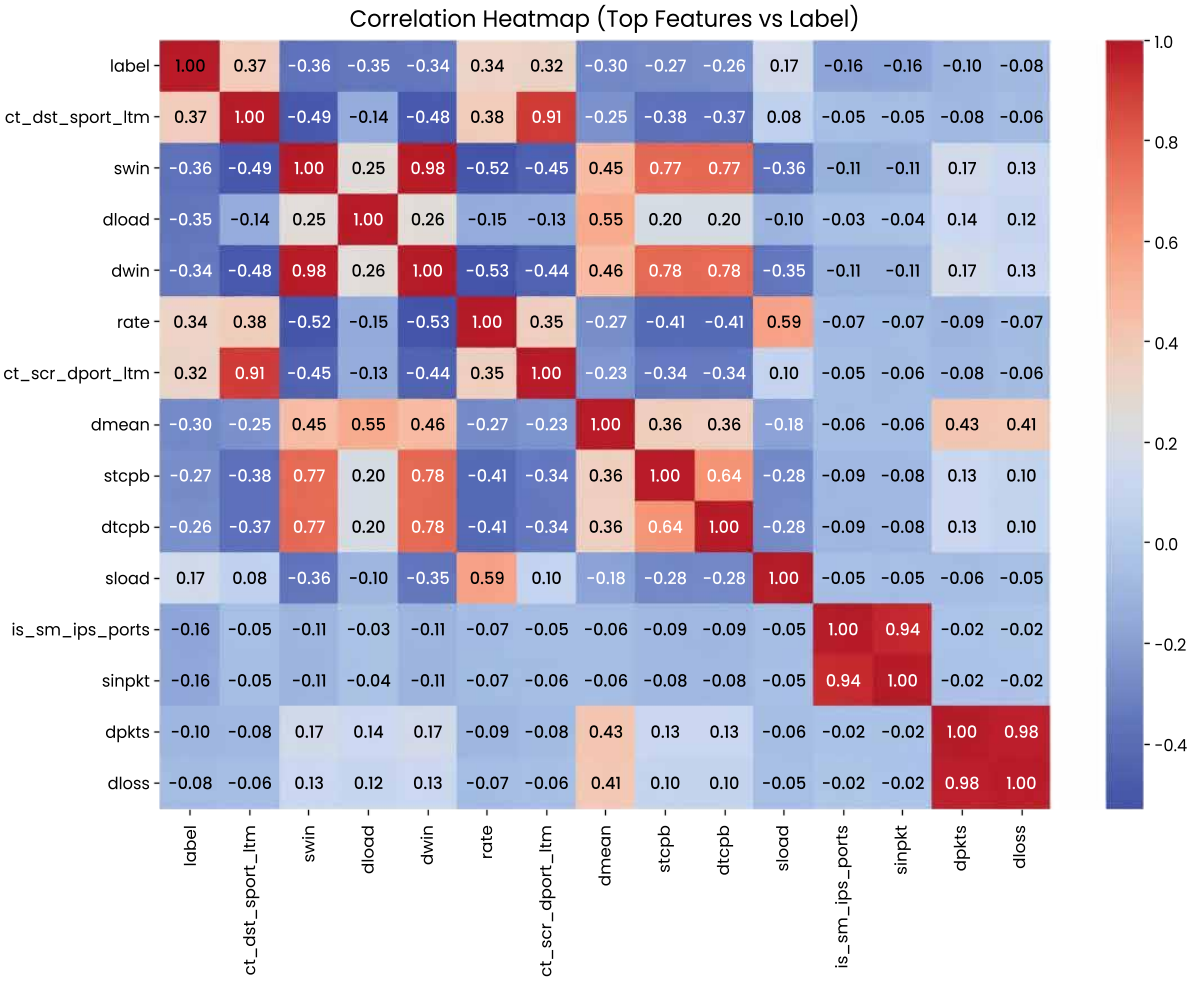

Fig. (5). Correlation heatmap (top features *vs* label).

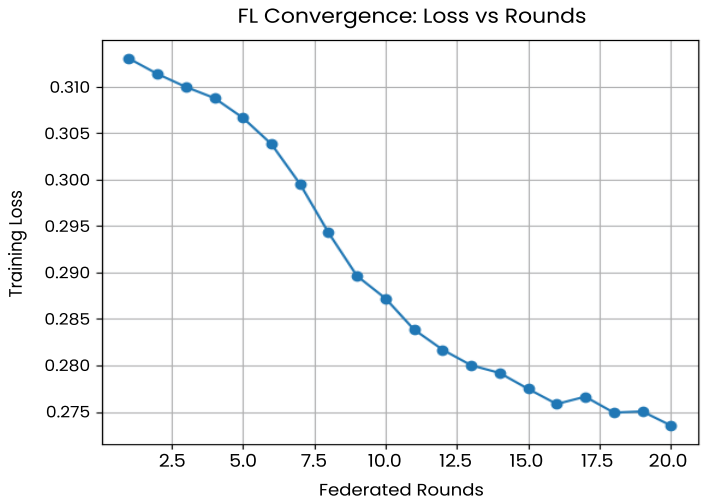

Fig. (6). FL Loss *vs* Rounds.

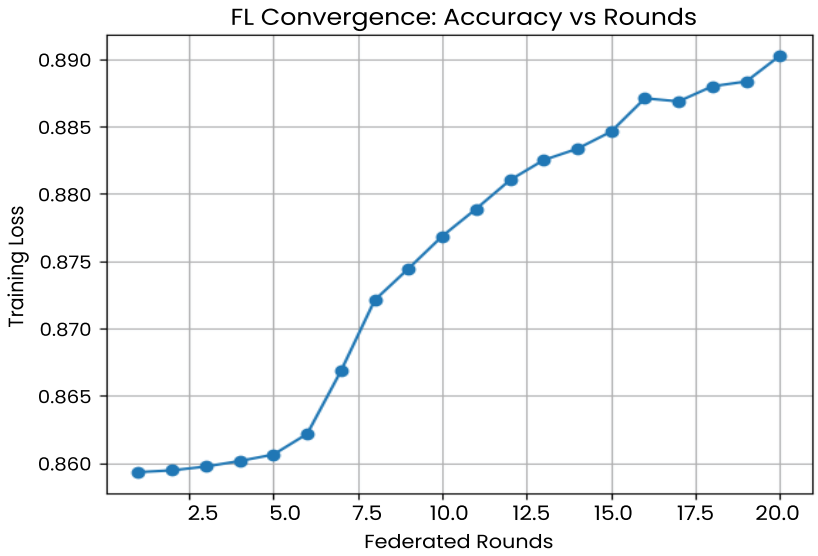

Fig. (7). FL Accuracy *vs* Rounds.

Table 3. Federated learning convergence (per round summary).

Metric	Initial Round	Final Round (20)
Training Loss	0.3130	0.2740
Training Accuracy	0.8590	0.8900

These convergence results show that Federated Averaging (FedAvg) remains effective for intrusion detection under heterogeneous non-IID client data. This result directly answers a frequent issue in federated learning literature and can prove the feasibility of FL in real-world scenarios, in the context of cybersecurity [18, 19, 24].

4.3. Client-Side Computational Performance

The time of client-side computation is a valuable indicator of the feasibility of federated deployment. The local training time taken by each client on averaging a communication round is shown in Fig. (**8**), with summary statistics presented in Table **4**. The average local training time is less than five seconds across all clients, and the minimum and maximum values are relatively similar to each other.

This consistency shows that the deep neural network structure and training setup are equally balanced in computational load, even though the local dataset sizes and composition are different. The fact that the proposed training framework can potentially be run on commodity hardware or edge-level infrastructure at a relatively low client-side cost indicates that the training framework does not place undue computational requirements on it [18, 24, 26]. This is one of the fundamental conditions of real-world implementation, especially in settings where the participating organizations might lack processing facilities or have severe operational limitations [10, 12, 18].

Table 4. Client-side training time per federated round.

Client ID	Training Time (s)
Client 0	4.98
Client 1	3.99
Client 2	4.10
Client 3	5.80
Client 4	3.93
Average	4.55
Minimum	3.93
Maximum	5.80

4.4. Comparative Model Performance

A central objective of this research is to assess the trade-offs between the predictive performance and preservation of privacy. Table **5** summarizes the comparative analysis between federated, centralized, local-only, and non-neural baseline models and is visually compared in Fig. (**9**).

The deep neural network is highly predictive due to its centralized nature and the complete visibility of the training data. Its accuracy, F1-score and area under the ROC curve are an upper-bound case where privacy is not considered. The Random Forest classifier achieves the best overall accuracy and F1-score among the considered approaches, indicating the efficiency of ensemble-based approaches for tabular intrusion detection when central access to data is available.

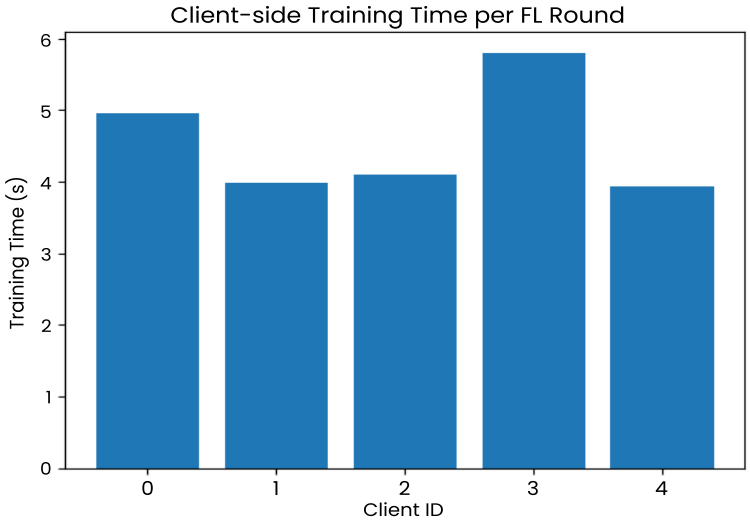

Fig. (8). Client training time per FL round.

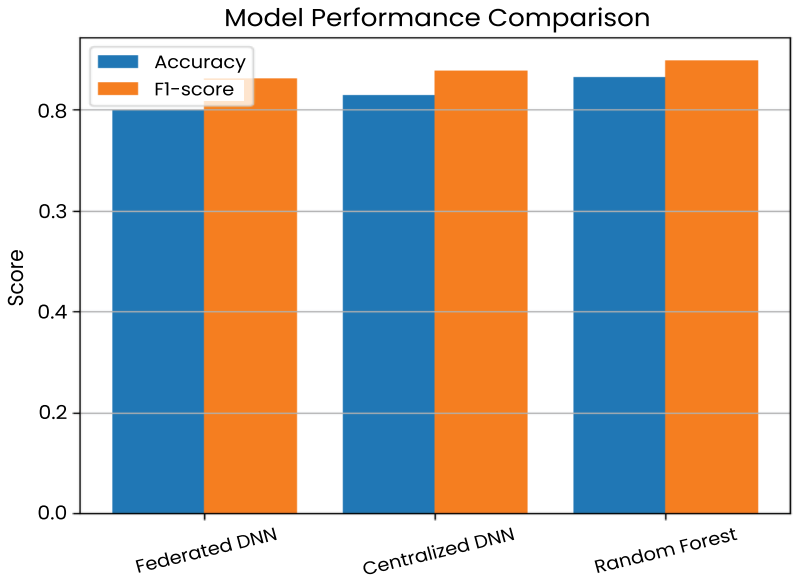

Fig. (9). Model performance comparison.

Table 5. Model performance comparison.

Model	Accuracy	Precision	Recall	F1-Score	AUC
Federated DNN	0.8019	0.7749	0.9725	0.8625	0.9060
Centralised DNN	0.8297	0.8109	0.9566	0.8777	0.9064
Random Forest	0.8646	0.8659	0.9325	0.8980	0.9154
Local-only DNN (avg)	0.7481	-	-	-	-

The federated deep neural network has a slightly lower accuracy when compared to its centralized counterpart, but has a high recall and AUC, which denotes high potential in identifying malicious activity. High recall value is especially important in intrusion detection, where the risk of a missed attack can be high. Notably, the federated model significantly outperforms the local-only deep neural networks, which are characterized by poor visibility and poor learning. This finding validates that federated learning through federated aggregation is vital in capturing global threat patterns in distributed environments [18, 27, 28].

The relatively small performance gap between federated and centralized learning demonstrates the practicality of the proposed framework. This trade-off is acceptable, considering that the federated approach does not require the sharing of raw data, which is likely to be unacceptable in privacy-sensitive environments [10, 22, 26]. Confidence intervals and statistical significance testing were not performed in this study, which constitutes a limitation of the comparative evaluation.

4.5. System-Level Overhead and Communication Cost

Federated learning has the benefits of privacy, although it comes with new system-level overhead relative to centralized training. Table **6** summarizes the time and costs of training and communication of all the approaches evaluated. The overall time spent in training the federated one is more than the time spent in centralized methods, since there is a communication round and overhead in the synchronization.

Table 6. System-level training and communication cost.

Metric	Value
Federated training time	449.3000 s
Centralized DNN training time	183.6900 s
Random Forest training time	160.4500 s
Total FL communication	10,854,800.0000 bytes (~10.85 MB)

Nevertheless, the absolute cost of communication is small, and the amount of data transferred is restricted to the model parameters, not raw traffic logs [24, 26]. The communication volume that was measured (in the range of several megabytes) is insignificant in comparison with the volumes of network traffic and is quite within the reach of contemporary enterprise networks [12, 18]. This finding indicates that the suggested framework is scalable and does not have out-of-bandwidth demands. This system-level evaluation differentiates the proposed framework from much of the existing FL-IDS literature, which often focuses on detection accuracy alone without analysing communication cost or round efficiency.

The extra training time that federated learning takes is a fair and justified trade-off when taken together with the privacy advantages [18, 20, 26]. Security model training is often conducted offline or periodically in many operating environments, and thus, small increases in training time are acceptable.

4.6. Privacy–Utility Trade-Off Analysis

The proposed framework is privacy-aware through strict data locality. The federated approach would greatly minimize the chances of direct data exposure and regulatory non-compliance by making sure that raw network traffic data never exits the client environment [10, 18, 24]. Table **7** summarizes the privacy settings and the differential privacy settings applied in the experiments.

Table 7. Privacy characteristics of the federated framework.

Parameter	Value
Privacy mechanism	Data locality + model aggregation
Raw data sharing	None
Noise injection	None
Secure aggregation	Not implemented
Formal DP guarantee	Not provided

Even though better formal privacy guarantees might be obtained by explicit noise injection, the findings indicate that meaningful threat detection performance may be obtained mainly due to the locality of data [20]. The observed performance degradation relative to centralised learning reflects a trade-off between communication efficiency and detection accuracy, rather than formal privacy noise, since no differential privacy mechanisms are employed.

DISCUSSION AND IMPLICATIONS

The findings show that federated learning is a practical and effective approach to privacy-aware threat detection in distributed IT systems. Table **8** contains a condensed summary

of the most important findings. The suggested framework can achieve a balance between the performance of detection, the efficiency of the system and privacy concerns in the real-life non-IID situation [18, 19]. Even though centralized models are slightly better in raw accuracy, the fact that they depend on data aggregation makes them inapplicable in most real-world situations [10-12, 22].

Although UNSW-NB15 contains a wide variety of attack categories and rich flow-level features, it remains a synthetic, lab-generated dataset with fixed attack distributions. Binary label collapse further removes operational nuance. Therefore, results may not fully generalise to live production environments.

A key limitation of this study is its reliance on a single benchmark dataset. While UNSW-NB15 is a widely used and well-regarded evaluation benchmark for intrusion detection, the generalisability of the findings to other network environments, traffic distributions, and attack taxonomies cannot be assumed without cross-dataset validation. Datasets such as CIC-IDS2017, CIC-IDS2018, and NSL-KDD capture different traffic profiles and attack scenarios; evaluating the proposed federated framework on such datasets is an important direction for future work to confirm the robustness and transferability of these results.

The results had immediate implications for collaborative security analytics across organizational boundaries, *e.g.,* managed security service provision or threat intelligence sharing within a sector. Federated learning provides a viable way forward to collective defense in more and more interconnected digital ecosystems by allowing cooperative learning without revealing sensitive data [18, 24, 27].

Table 8. Summary of key experimental findings.

Aspect	Observation
Data distribution	Highly skewed, non-IID
FL convergence	Stable within 15–20 rounds
Privacy–utility trade-off	Minor accuracy loss vs centralized
Best accuracy	Random Forest (centralized)
Best privacy-preserving model	Federated DNN

While federated learning reduces privacy risks by retaining data locally, this work does not provide formal differential privacy guarantees and remains vulnerable to potential gradient leakage attacks. No secure aggregation or differential privacy mechanisms are employed. Therefore, privacy preservation in this study should be interpreted as privacy-aware rather than provably private.

It is important to note that even under a data-local federated learning regime, meaningful privacy risks persist. Specifically, gradient inversion attacks can partially reconstruct training data from model updates shared with the server, and membership inference attacks can determine whether a specific sample was used during training. These attack vectors highlight the gap between data locality and formal privacy protection. Future extensions of this framework should integrate differential privacy mechanisms such as Gaussian noise injection with calibrated privacy budgets and secure aggregation protocols, which would provide measurable and provable privacy guarantees beyond the data-locality assurances offered by the current implementation.

CONCLUSION

This paper analysed the feasibility and efficiency of federated learning as a privacy-aware threat detection mechanism in distributed IT systems. In response to the weaknesses of centralized intrusion detection architectures and the growing limitations of privacy policies and organizational data silos, the proposed work empirically tested an end-to-end federated learning model with the UNSW-NB15 dataset. The framework allowed collaborative learning of threat intelligence by incorporating deep neural networks into a federated optimization framework, whereby the underlying network traffic data did not leave their respective entities.

The experimental findings showed that the proposed federated deep neural network had high predictive performance in realistic conditions of non-IID data. Despite the marginally higher overall accuracy of the centralized learning method compared to the federated method, the federated method retained a high recall and area under the ROC curve, which is the most important measure in intrusion detection conditions when false negatives are potentially disastrous. Notably, the federated model was far more effective than local-only learning, which supports the notion that cooperative combination of learned representations is necessary in order to represent the global threat patterns in a distributed environment.

Besides predictive accuracy, the paper also introduced a system-level analysis, including convergence behaviour, client-side computational price, training time, communication overhead, and privacy. The federated model demonstrated consistent convergence with a small number of communication rounds regardless of the heterogeneous data distributions of clients. Computation times were consistent and low on the client-side, meaning that the framework was computationally viable to be deployed on commodity hardware or edge-level infrastructure. Also, the communication overhead was seen to be modest and significantly lower than that which might be needed to compute data aggregation centrally, which supports the scalability of the proposed solution.

Privacy-wise, the framework reduced data exposure by ensuring that data locality is observed, hence limiting regulatory and security risks of centralized data collection. Although no formal differential privacy mechanisms were implemented, the results show that federated collaboration can still achieve meaningful threat detection performance.

A number of limitations were identified in this study, which informed several directions for further investigation. Potential improvements included the integration of secure aggregation mechanisms and the incorporation of stronger formal privacy guarantees, such as adaptive differential privacy techniques.

Extensions to support streaming data, concept drift handling, and fine-grained multi-class attack classification were also identified as important steps toward enhancing practical applicability. Moreover, the framework was identified as a candidate for evaluation in real multi-organizational deployment settings in order to further assess its robustness, scalability, and real-world utility.

LIST OF ABBREVIATIONS

DP	=	Differential Privacy
DNNs	=	Deep Neural Networks
FL	=	Federated Learning
GDPR	=	General Data Protection Regulation
IDS	=	Intrusion Detection System
ML	=	Machine Learning

AUTHOR'S CONTRIBUTION

Im.U. has contributed to study concept and design, data collection and data analysis. Ib.U., N.U. has contributed in writing the paper and results interpretation.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data will be made available on reasonable request by contacting the corresponding author [Im.U.].

FUNDING

None.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this manuscript. No financial, personal, or professional relationships have influenced the content or outcomes of this study.

ACKNOWLEDGEMENTS

The author would like to express their sincere gratitude to all the researchers whose work was reviewed and cited in this study. Their contributions have been instrumental in shaping the insights and findings of this systematic literature review. We also thank the academic community and institutions providing access to high-quality digital libraries and databases, which enabled a comprehensive and rigorous analysis.

DECLARATION OF AI

During the preparation of this work the author used ChatGPT for editing purposes. After using this tool, the author reviewed and edited the content as needed and take full responsibility for the content of the published article.

REFERENCES

[1] Albshaier L, Almarri S, Albuali A. Federated learning for cloud and edge security: A systematic review of challenges and AI opportunities. Electronics 2025; 14(5): 1019. https://doi.org/10.3390/electronics14051019

[2] Chen C, *et al.* Trustworthy federated learning: privacy, security, and beyond. Knowl Inf Syst 2025; 67(3): 2321-2356. https://doi.org/10.1007/s10115-024-02285-2

[3] Bilal G, Meriem B. Federated learning for cybersecurity: Enhancing threat detection across multiple organizations [Thesis]. University of Biskra; 2025. Available from: http://archives.univ-biskra.dz/bitstream/123456789/31490/1/Boussaha_Meriem_Ghamri_Bilal.pdf

[4] Samuel AJ. Optimizing energy consumption through AI and cloud analytics: Addressing data privacy and security concerns. World J Adv Eng Technol Sci 2024; 13(2): 789-806. https://doi.org/10.30574/wjaets.2024.13.2.0609

[5] Ullah I, Yaseen MU, Amin NU, Qureshi MR, Ibrahim S. Explainable emotion recognition from heart rate data using deep learning and XGBoost. In: Proc 2025 27th Int Multitopic Conf (INMIC). IEEE; 2025. https://doi.org/10.1109/INMIC65900.2025.11348573

[6] Amin, R., Costanzo, A., Alzabin, L.R. *et al.* An efficient federated learning-based defense mechanism for software defined network cyber threats through machine learning models. Sci Rep 2025; 15(1): 41390. https://doi.org/10.1038/s41598-025-25345-1

[7] Vyas A, Lin PC, Hwang RH, Tripathi M. Privacy-preserving federated learning for intrusion detection in IoT environments: a survey. IEEE Access. 2024; 12: 127018-127050. https://doi.org/10.1109/ACCESS.2024.3454211

[8] Siddiqui MR. Big data and great privacy challenges in the digital era-A comprehensive study. Innov J Appl Sci 2025; 2(6): 42. https://doi.org/10.70844/ijas.2025.2.42

[9] Rahim T, Ullah I, Nazir A, Tanveer MS, Qureshi MR. A deep learning approach to PCOS diagnosis: Two-stream CNN with transformer attention mechanism. Spectr Eng Sci 2025; 3(7). https://doi.org/10.5281/zenodo.15790016

[10] Liu Y, James J, Kang J, Niyato D, Zhang S. Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet Things J 2020; 7(8): 7751-7763. https://doi.org/10.1109/JIOT.2020.2991401

[11] Hasan MM. Federated learning models for privacy-preserving AI in enterprise decision systems. Int J Bus Econ Insights 2025; 5(3): 238-269. https://doi.org/10.63125/ry033286

[12] Gadekallu TR, *et al.* Federated learning for big data: A survey on opportunities, applications, and future directions. 2021. https://doi.org/10.48550/arXiv.2110.04160

[13] Raza M, Saeed MJ, Riaz MB, Sattar MA. Federated learning for privacy-preserving intrusion detection in software-defined networks. IEEE Access. 2024; 12: 69551-69567. https://doi.org/10.1109/ACCESS.2024.3395997

[14] Quffa A, Abu-Naser SS. A rule-based expert system for cybersecurity threat detection: Evolution, applications, and the hybrid AI paradigm. Int J Acad Eng Res 2025; 9(8): 44-62 Available from: http://ijeais.org/wp-content/uploads/2025/8/IJAER250807.pdf

[15] Hozouri A, Mirzaei A, Effatparvar M. A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. Discov Artif Intell 2025; 5(1): 314. https://doi.org/10.1007/s44163-025-00578-1

[16] Sharif F. The role of ensemble learning in strengthening intrusion detection systems: A machine learning perspective. Int J Comput Eng Technol 2024. Available from: https://www.researchgate.net/publication/384366905_The_Role_of_Ensemble_Learning_in_Strengthening_Intrusion_Detection_Systems_A_Machine_Learning_Perspective

[17] Latif N, Ma W, Ahmad HB. Advancements in securing federated learning with IDS: a comprehensive review of neural networks and feature engineering techniques for malicious client detection. Artif Intell Rev 2025; 58(3): 91. https://doi.org/10.1007/s10462-024-11082-w

[18] Alketbi KS, Mehmood A. A comprehensive survey of explainable artificial intelligence techniques for malicious insider threat detection. IEEE Access 2025; 13: 121772-121798. https://doi.org/10.1109/ACCESS.2025.3587114

[19] Chowdhury TK. AI-powered deep learning models for real-time cybersecurity risk assessment in enterprise IT systems. ASRC Procedia Glob Perspect Sci Scholarsh 2025; 1(01): 675-704. https://doi.org/10.63125/137k6y79

[20] Li W, Meng W, Kwok LF. Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions. IEEE Commun Surv Tutor 2021; 24(1): 280-305. https://doi.org/10.1109/COMST.2021.3139052

[21] Mankotia S, de Leon DC, Rimal BP. FedPrIDS: privacy-preserving federated learning for collaborative network intrusion detection in IoT. J Cybersecur Priv. 2026; 6(1): 10. https://doi.org/10.3390/jcp6010010

[22] Khalil U, Malik OA, Uddin M, Chen CL. A comparative analysis on blockchain versus centralized authentication architectures for IoT-enabled smart devices in smart cities: a comprehensive review, recent advances, and future research directions. Sensors 2022; 22(14): 5168. https://doi.org/10.3390/s22145168

[23] Alqattan DSM. Security of distributed and federated deep learning systems [thesis]. Newcastle University; 2025. http://hdl.handle.net/10443/6614

[24] Ji, S, Tan, Y, Saravirta, T. *et al*. Emerging trends in federated learning: From model fusion to federated x learning. Int J Mach Learn Cybern 2024; 15(9): 3769-3790. https://doi.org/10.1007/s13042-024-02119-1

[25] Zhao Z, *et al.* Federated learning with non-IID data in wireless networks. IEEE Trans Wirel Commun 2021; 21(3): 1927-1942. https://doi.org/10.1109/TWC.2021.3108197

[26] Bouacida N, Mohapatra P. Vulnerabilities in federated learning. IEEE Access 2021; 9: 63229-63249. https://doi.org/10.1109/ACCESS.2021.3075203

[27] Taheri R, Jafari R, Gegov A, Arabikhan F, Ichtev A. Explainable AI for federated learning-based intrusion detection systems in connected vehicles. Electronics 2025; 14(22): 4508. https://doi.org/10.3390/electronics14224508

[28] Baich M, Sael N. A federated learning-based intrusion detection system using dynamic ensemble aggregation for IoT networks. IEEE Access 2025; 13: 205826-205839. https://doi.org/10.1109/ACCESS.2025.3640521

[29] Hernandez-Ramos JL, *et al.* Intrusion detection based on federated learning: A systematic review. ACM Comput Surv 2025; 57(12): 1-65. https://doi.org/10.1145/3731596

[30] Albanbay N, *et al.* Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study. J Sens Actuator Netw 2025; 14(4): 78. https://doi.org/10.3390/jsan14040078

[31] Moualla S, Khorzom K, Jafar A. Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset. Comput Intell Neurosci 2021; 2021(1): 5557577. https://doi.org/10.1155/2021/5557577

[32] Jouhari M, Benaddi H, Ibrahimi K. Efficient intrusion detection: Combining x2 feature selection with CNN-BiLSTM on the UNSW-NB15 dataset. Proc 2024 11th Int Conf Wirel Netw Mob Commun (WINCOM). IEEE 2024: 1-6. https://doi.org/10.1109/WINCOM62286.2024.10658099

[33] Pinheiro JMH, *et al.* The impact of feature scaling in machine learning: Effects on regression and classification tasks. IEEE Access 2025; 13: 199903-199931. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=11261543

[34] Shebl A, Elsedimy EI, Ismail A, Salama AA, Herajy M. DCNN: A novel binary and multi-class network intrusion detection model via deep convolutional neural network. EURASIP J Inf Secur 2024; 2024(1): 36. https://doi.org/10.1186/s13635-024-00184-1

[35] Efthymiadis F, Karras A, Karras C, Sioutas S. Advanced optimization techniques for federated learning on non-IID data. Future Internet 2024; 16(10): 370. https://doi.org/10.3390/fi16100370