ITUJournal
Future and evolving technologies
ITU Journal on Future and Evolving Technologies, Volume 7, Issue 2, June 2026
ARTICLE

FLAccShield: Federated learning with accuracy-weighted aggregation for DDoS detection

Md Mahibul Hasan[1], Aymen Ben Said[1], Karamveer Singh Sidhu[2], Nashid Shahriar[1], Sajal Saha[2]

[1] Department of Computer Science, University of Regina, 3737 Wascana Pkwy, Regina, S4S0A2, Saskatchewan, Canada, [2] Department of Computer Science, University of Northern British Columbia, 3333 University Way, Prince George, V2N 4Z9, British Columbia, Canada

Corresponding author: Nashid Shahriar, nashid.shahriar@uregina.ca

5G network slicing enables flexible and efficient service deployment by maintaining logical isolation and facilitating tailored Quality of Service (QoS) and security policies. However, network slices increase network complexity and expand the attack surface, making them more susceptible to Distributed Denial-of-Service (DDoS) attacks. Due to the growing attack landscape, Machine Learning (ML)-based Intrusion Detection Systems (IDS) are becoming more popular over traditional IDS. Since network slices are used by different verticals, they often have strict privacy requirements. Conventional ML-based IDS are centralized in nature, combining all data on a central server, which fails to uphold slice-level privacy constraints. This paper proposes FLAccShield, a Federated Learning (FL)-based privacy-preserving IDS framework for DDoS detection in 5G networks through collaboration among multiple slices. The framework is rigorously evaluated for supervised and unsupervised settings for both Independent and Identically-Distributed (IID) and non-IID datasets against a varied range of features. The experimental results demonstrate that FLAccShield achieves consistent and stable detection performance across varying feature set sizes and data distributions. In supervised settings with non-IID data, the most practically challenging scenario, FLAccShield outperforms FedAvg and FedProx by approximately 3% in F1 score while exhibiting substantially less performance degradation as feature count increases (~7% decline versus ~10–22% for baselines). In supervised IID settings, FLAccShield maintains uniform per-client performance where baselines show client-level deviation. In unsupervised anomaly detection, FLAccShield achieves comparable performance to FedAvg and FedProx (within 0.4–1%), while QFedAvg proves poorly suited to the autoencoder-based anomaly detection paradigm. These findings highlight FLAccShield's robustness and its ability to preserve privacy in complex 5G environments.

Keywords: 5G network slice, distributed denial of service (DDoS), federated learning (FL), intrusion detection system (IDS), privacy-preserving security

1. INTRODUCTION

The fifth generation mobile networks are reshaping the future of telecommunications through Network-as-a-Service (NaaS). By enabling customizable, on-demand connectivity, 5G supports a broad spectrum of applications such as mission-critical systems, autonomous driving, Industry 4.0, extended Reality (XR), Internet of Things (IoT), and remote healthcare [1]. Each of these verticals has distinct requirements for latency, bandwidth, reliability, and security. Meeting these diverse requirements calls for dedicated, logically-isolated resources that can be tailored to ensure QoS and data privacy. Network slice is a key feature in 5G architecture that addresses these challenges by creating multiple virtual networks on a shared physical infrastructure. Each slice is optimized for a specific service and QoS. For example, slices that support autonomous vehicles require ultra-low latency and high reliability, while those for augmented Reality or video streaming prioritize high throughput. These slices operate independently, with full isolation, to avoid cross-slice interference and ensure user privacy. However, the

same flexibility and dynamic configuration that make network slicing powerful also introduce new security vulnerabilities. DDoS attacks remain a major concern, as they can target individual slices and consume critical resources [2]. Zero-day attacks pose an even greater risk, exploiting unknown vulnerabilities without any prior signatures.

Due to the increasing attack landscapes, traditional Intrusion Detection Systems (IDS) are no longer sufficient to offer comprehensive protection for 5G services and network slices. More adaptable, flexible and intelligent approaches like Machine Learning (ML)-based IDS are gaining popularity [3]. Although ML-based IDSs show great potential in identifying sophisticated threats, most current implementations use centralized architectures that require raw data aggregation. In a 5G network slice context, such centralization violates privacy and security policies, as data from multiple verticals is not allowed to be shared or moved outside their trusted boundaries. Therefore, privacy-preserving ML-based IDSs are better suited to uphold the integrity of slice-level operations in 5G environments.

In recent years, a considerable number of studies have explored Federated Learning (FL) as a privacy-preserving approach to IDS. FL enables collaborative model training without sharing raw data, making it well-aligned with the privacy demands of 5G slices. However, most of the existing FL-based IDS research primarily focuses on model training while overlooking privacy concerns during data generation and evaluation phases. Very few studies ensure end-to-end privacy across the entire pipeline (Data generation → Training → Evaluation), which is essential for real-world deployment. Moreover, since data remains local in FL setups, it opens opportunities for slice-level collaboration across distributed 5G testbeds. Table 1 summarizes recent work in FL-based IDS and highlights that very few approaches address both inter-slice collaboration and complete end-to-end privacy assurance.

To address these gaps, this paper presents FLAccShield, an FL-based IDS framework designed for privacy-preserving DDoS detection in 5G network slices. FLAccShield enables distributed model training across slices without requiring any raw data sharing. Each slice maintains full control over its local data while still contributing to a global model, and consequently preserves both the privacy and the security of the local data. Specifically, this paper makes the following key contributions:

- Introduces FLAccShield, a novel aggregation strategy that employs an adaptive learning rate scheduler and early stopping mechanisms at both local and global levels to reduce unnecessary computation and ensures stable convergence.
- Evaluates FLAccShield under both supervised classification (CNN, for attack types present in the training data) and unsupervised anomaly detection (autoencoder, for attack samples entirely absent from the training data) paradigms, across IID and non-IID data distributions. This dual-paradigm evaluation assesses the framework's adaptability across varying data heterogeneity scenarios and its ability to generalize to out-of-distribution traffic without requiring labeled attack samples.
- Shows how the models' performance varies in practice depending on the feature engineering techniques and the number of features used.

The remainder of this paper is structured as follows: Section 2 reviews existing literature on intrusion detection and FL in 5G networks. Section 3 outlines the proposed methodology, including dataset generation, feature selection, client setup, and model training strategy. Section 4 outlines the algorithm, deep learning models, and hyperparameters used to implement FLAccShield. Section 5 presents evaluation metrics, and results. Finally, Section 6 concludes the paper and outlines directions for future work.

2. RELATED WORK

In this section, we explore state-of-the-art research contributions related to IDS and their evolution, with a specific emphasis on the role of FL in enhancing security mechanisms in 5G networks. Traditional ML-based IDS approaches are deployed in a centralized manner which may not be suitable for modern decentralized and privacy-sensitive environments. These IDS models trained on unified datasets were effective in detecting anomalies in previous generation networks, but they struggle to scale in the data-intensive and latency-sensitive 5G architecture [9]. Additionally, these centralized systems are susceptible to single points of failure and raise concerns about raw data transmission, especially in privacy-sensitive applications [10]. A major limitation of a centralized system in multi-tenant 5G deployments is data privacy: aggregating raw traffic from multiple slice tenants onto a central server violates the data residency and confidentiality requirements that govern inter-tenant data sharing, particularly where slices belong to distinct legal entities with separate contractual obligations [10]. We note that federated learning addresses this privacy constraint specifically, by ensuring raw traffic never leaves its originating slice, but does not eliminate all architectural risks. The FL aggregation server remains a point of trust, and the distributed training process introduces new attack vectors, including model poisoning by malicious or malfunctioning clients. Moreover, the vast data generated by 5G networks strains the scalability of centralized solutions, often leading to increased latency and inefficiencies. Several pieces of work have explored

Table 1 – Summary of related work in FL IDS for 5G networks

Related Work	Dataset Used	5G Slice Available	Dataset Generation	Training	Evaluation	Inter-testbed Collaboration
[2]	DoS/DDoS Dataset [4]	✓	✓			
[5]	Not published		✓	✓	✓	
[6]	CIC-DDoS2019 [7]					
[8]	DoS/DDoS Dataset [4]	✓	✓			
This Work	DoS/DDoS Dataset [4]	✓	✓	✓	✓	✓

non-federated IDS systems tailored for 5G environments. Khan et al. proposes a centralized deep learning–based intrusion detection system for detecting DoS/DDoS attacks on 5G network slices using a BiLSTM model trained on traffic collected from a simulated Free5GC and UERANSIM testbed [2]. In their subsequent thesis work, the authors further evaluated both BiLSTM and CNN architectures for slice-level attack detection under centralized training settings [11]. While these approaches demonstrated high detection accuracy (99.99%), they rely on centralized data aggregation and therefore preserve privacy only up to the data generation stage. In contrast, our work adopts the CNN architecture proposed in [11] and extends it within a federated learning framework, enabling decentralized and privacy-preserving collaborative intrusion detection across slice-isolated domains. Authors of [8] construct slice-wise datasets by centrally aggregating and mixing traffic data, followed by flow-level sampling to emulate network slices, and apply deep learning–based DDoS detection under centralized training settings. While this approach is effective for evaluation, the centralized data preparation and training pipeline does not preserve data privacy. In contrast, our work avoids centralized data aggregation and enables privacy-preserving intrusion detection by adopting federated learning across slice-isolated domains within the 5G core. The paper by Silva et al. [12] focuses on the significant issue of DDoS attacks on the 5G control plane, which can compromise service availability and security. The authors propose a new approach, REPEL, an intelligent resource-scheduling strategy that utilizes game theory to combat these attacks. The study examines the efficiency and effectiveness of REPEL through a queuing model and an experimental testbed utilizing a virtualized evolved packet-core prototype. Chilukuri et al. [13] present the SENTINEL framework that utilizes the Self-Organizing Network (SON) paradigm and learning based on Hierarchical Temporal Memory (HTM) to protect the 5G core control plane from DDoS attacks. The framework detects and isolates malicious users attempting to attack the 5G core using a slice aggregator and Multi-Factor Authentication (MFA). The efficacy of SENTINEL is evaluated through experimentation on a 5G testbed and utilization of a semi-synthetic dataset of anomalies. The results indicate that SENTINEL maintains high levels of service availability for legitimate users while avoiding the expenditure of additional resources. Serrano Mamolar et al. [14] provide an autonomous security method in multi-tenant 5G networks that can effectively identify DDoS attacks without human involvement. The approach employs a linked self-regulating control loop consisting of a Security Monitoring Agent (SMA), decision maker, action enforcer, and flow control agent, each with a distinct purpose in identifying, detecting, and mitigating DDoS attacks. The experiments demonstrate that the suggested approach can manage and mitigate attacks, even when confronted with 256 attackers concurrently. Whitworth et al. [15] develop a new DL-based method known as DeepSecure that utilizes an LSTM model to distinguish between legitimate and malicious network traffic. To achieve this, it analyzes network traffic characteristics and predicts whether they indicate normal behavior or a DDoS attack on 5G network slicing. After confirming the traffic is legitimate, DeepSecure accurately predicts the most appropriate network slice based on the required service type. The results show a detection accuracy of 99.97% and a slice prediction accuracy of 98.798%, surpassing the performance of conventional ML and DL methods used in prior research. Djuitcheu et al. [16] comprehensively analyzes the impact of DDoS attacks on important 5G network performance measures, including latency, jitter, packet loss, and bit rate. To achieve this, a controlled 5G testbed environment was employed to generate and monitor DDoS attacks and their impact on network performance. Tools such as hping3 [17] and LOIC [18] were used to simulate DDoS attacks, while iperf3 and ping were used to evaluate network performance. The results showed that DDoS attacks substantially impacted latency and jitter, leading to a decrease in bit rate and an increase in packet loss. Gao et al. [19] propose the G/M/1 queuing model as a new edge server job scheduling approach in order to mitigate DDoS attacks based on 5G networks. The main goal is to improve load distribution and minimize the effects of DDoS attacks by maximizing users' Quality of Experience (QoE). It utilizes a nonstationary Multi-Armed Bandit (MAB) algorithm framework to effectively distribute the workload among edge nodes. An actual dataset is used to measure the efficacy of the proposed method. Agrawal et al. [20] present a thorough investigation of FL in the context of intrusion detection systems. By training models locally on devices and communicating just the model parameters to a central server, FL, as a decentralized learning approach, protects security and improves privacy. The authors in [20] investigate numerous technologies and methods that use ML, DL, and FL for intrusion detection, highlighting their advantages and synergies. Additionally, they emphasized current difficulties in the use of FL in intrusion detection and

suggest potential future study routes, successfully laying the groundwork for greater research in this field. This thorough study offers a useful insight of FL's function and possibilities for enhancing intrusion detection systems. With a motivation to leverage immense capabilities of FL described in [20], we have employed FL as a part of our solution to tackle DDoS attack and protect decentralized 5G network in a slicng environment. FL has been applied to different network challenges, especially those focused on privacy [21] [22] [23]. Early frameworks such as the FedAvg algorithm established the foundation of FL by optimizing communication efficiency during model training [24]. These advancements have positioned FL as a pivotal tool in privacy-preserving distributed ML. FL has shown significant promise in enhancing IDS by enabling decentralized anomaly detection. For instance, DeepFed, a federated framework combining Gated Recurrent Units (GRUs) and CNNs, was introduced to detect DDoS attacks with high accuracy [10]. Moreover, in recent days, FL-based IDS frameworks are demonstrating remarkable success in detecting complex threats while preserving data privacy [25] [26]. Techniques like adaptive aggregation and client selection have further enhanced model performance in non-IID data scenarios. Studies showcasing these advancements validate FL's ability to maintain high detection accuracy and scalability in distributed environments [9]. Marfo et al. [27] propose Hierarchical FL architecture consisting of three layers which are clients, edge servers and global servers. Fully-connected Artificial Neural Networks (ANNs) with 2 hidden layers populating 256 perceptron in each layer has been used as local and global models. One of the significant findings of this research is the reduced training time advantage of FL in comparison to centralized models. The trade-offs between communication expense and training time based on frequency of contact with the servers were underlined by the authors in this paper. To conduct experiments, researchers have used the UNSW-NB15 dataset [28] to feed to their model, which is basically constructed out of synthetic environment along with major issues such as class imbalance and class overlap [29]. In [5], the authors proposed a federated intrusion detection framework and evaluated its performance under both IID and non-IID data distributions in a peer-to-peer federated learning setting. Their study primarily focused on distributed learning behavior and robustness in B5G Open RAN environments, without explicitly considering network slicing or slice-isolated traffic domains. In contrast, while we also evaluate IID and non-IID federated learning scenarios, our work explicitly incorporates 5G network slicing from the 5G core, modeling slices as isolated security domains and assessing privacy-preserving intrusion detection across heterogeneous slice-specific traffic. Maiga et al. [6] employ federated learning at the VNF level, treating individual 5G core network functions as FL clients for privacy-preserving DDoS detection. However, their framework does not consider network slicing or slice-isolated traffic domains. In contrast, our work models 5G network slices as federated entities, enabling slice-aware and privacy-preserving intrusion detection across heterogeneous slice-specific traffic. Boualouache et al. [30] present a solution for detecting inter-slice attacks in Vehicle to everything (V2X) 5G networks, which pose a significant threat to the isolation and privacy of network slices. The proposed approach utilizes federated and deep learning to deploy virtual security functions within network slices, which cooperate to train and refine attack detection models.

In summary, while previous research has made significant strides in FL-based IDS frameworks for 5G networks, several gaps remain (Table 1). These include insufficient privacy preservation during slice-level collaboration, unrealistic dataset generation methods, and the absence of evaluation under both supervised and unsupervised learning paradigms, particularly for detecting DDoS attack categories not represented in the training data. In this paper, we address these limitations by maintaining end-to-end privacy across the full pipeline from data generation through evaluation, using realistic slice-isolated dataset generation across heterogeneous 5G testbeds, and evaluating both supervised classification of known attack types and unsupervised anomaly detection of attack categories absent from the training distribution.

3. SYSTEM MODEL AND METHODOLOGY

3.1 Testbed architecture

A 5G network slice is a logically-isolated end-to-end virtual network instantiated over a shared physical infrastructure, with dedicated control and data plane functions tailored to a specific service vertical [1]. In the 3GPP architecture, slice isolation is enforced through dedicated instances of the Session Management Function (SMF), which handles session establishment and policy enforcement, and the User Plane Function (UPF), which anchors the data plane and serves as the traffic collection point for each slice. This architectural separation guarantees that packet flows belonging to one slice are never processed or observable by another, a property we exploit directly in our FL design.

We instantiate this architecture across two independent 5G testbeds to increase the volume and diversity of slice-specific training data available to the federated system. The first testbed deploys all network functions as Virtual Machines (VMs); the second uses containerized deployments of the same functions. Both testbeds share an identical logical topology and generate network flows at the same protocol level.

The 5G core network functions, including AMF, SMF, UPF, and supporting components, are simulated using

free5GC [31], an open-source 3GPP Release 15-compliant 5G core implementation. We note that free5GC, as a research-oriented platform, does not enforce slice isolation cryptographically or through a standards-hardened isolation mechanism. In our testbed, slice isolation is therefore achieved architecturally: we deploy entirely separate gNB, SMF, and UPF instances for each slice, with no shared network functions across slice boundaries. Under this configuration, traffic from one slice is physically routed exclusively through its designated UPF and is never processed by, or observable from, any other slice's functions.

Slice isolation is enforced by deploying separate gNB, SMF, and UPF instances for each slice. Critically, all user traffic is routed exclusively through the UPF of its corresponding slice, with no cross-slice data plane visibility. This makes the UPF the natural and privacy-preserving collection point for network traffic: all PCAP captures are performed at the UPF using `tcpdump` [32], ensuring that each FL client's training data reflects only the traffic within its own slice boundary and is never accessible to the aggregation server or to other clients in raw form.

Each network slice is modeled as an independent FL client. The mapping is direct: the slice boundary enforced at the UPF level corresponds exactly to the data locality constraint assumed by HFL, a client holds data it cannot share. The FL aggregation server operates outside all slice domains, receiving only model weight tensors and scalar performance metrics from each client. Fig. 1 illustrates this architecture, showing the correspondence between the 5G core slice structure and the FL client topology.

3.2 Dataset generation

Datasets play a crucial role in AI/ML-based solutions, especially when tailored to specific domains. Recent studies have introduced domain-specific datasets, such as CIC-IDS2017 and CIC-IDS2018, which were published for IDS purposes [33], CIC-DDoS2019, which contains extensive DDoS attack traffic [7], and the 5G-NIDD dataset, which was generated from a 5G environment [34]. However, publicly available datasets have their limitations and are not suitable for our experiments. Although CIC-IDS2017, CIC-IDS2018, and CIC-DDoS2019 contain a wide variety of attacks, they are not generated from a 5G environment. On the other hand, although the 5G-NIDD dataset is derived from a 5G environment, it lacks slice-level isolation, which is essential for accurate experimentation on 5G network slices. To address this gap, in our previous contribution, we released a DoS/DDoS attack dataset derived from two simulated 5G testbeds, each consisting of two distinct slices [4]. This dataset stands out as one of the few publicly available resources that maintains appropriate slice-level isolation within data plane traffic, ensuring a more realistic representation of real-world network segmentation.

For our experiment, we generated both benign and attack traffic from the UEs. To simulate benign Internet traffic, we utilized an automated Python script with a headless browser to navigate across 500 different websites. The script performed various activities, including streaming videos, transferring and copying files, downloading and modifying data, executing ICMP pings, conducting SSH operations, and other common Internet tasks. We use hping3 to generate eight distinct types of DDoS attacks: UDP flood, TCP SYN, TCP PUSH, TCP ACK, TCP FIN, TCP URG, TCP XMAS, and TCP YMAS. We collected all network traffic as Packet Capture (PCAP) files from UPF using tcpdump [32].

In our experiments, we employed flow-based features, requiring conversion of PCAP files into network flows. While CICFlowMeter was a viable tool, its official release [36] contained bugs that generated unreliable flow features. Elengen et al. addressed those bugs and proposed an improved version of CICFlowMeter [37] that we use for generating flow-based features. In our data generation process, we have introduced an extra preprocessing step where we verify the accuracy of key features by comparing them against wireshark [38] trace analysis to confirm their validity. During generation of flow data for each client, we set the maximum flow duration to 10 000 seconds in CICFlowMeter to ensure that no flows were split into multiple sub-flows, and preserve the integrity of long-duration network sessions. The entire data generation pipeline is shown in Fig. 2. We have made our dataset publicly available on IEEE Dataport [4].

The slice-level isolation enforced during collection means that no client can access another's raw traffic, directly motivating the federated training approach in FLAccShield where only model weights, not data, are shared across slice boundaries.

3.3 Dataset preparation and preprocessing

We have prepared two subsets using the complete dataset, each designed to evaluate different data distribution scenarios. In the IID scenario, eight out of the total attack types were uniformly distributed across all clients to make sure that each client had access to the same set of attacks. In contrast, the non-IID scenario has been structured to create a more heterogeneous distribution, where each client was assigned only two distinct attack types. Tables 2, and 3 illustrate the attack distributions of both of the scenarios. The dataset does not have separate sub-classes for benign traffic and its distribution remains the same in both IID and non-IID settings.

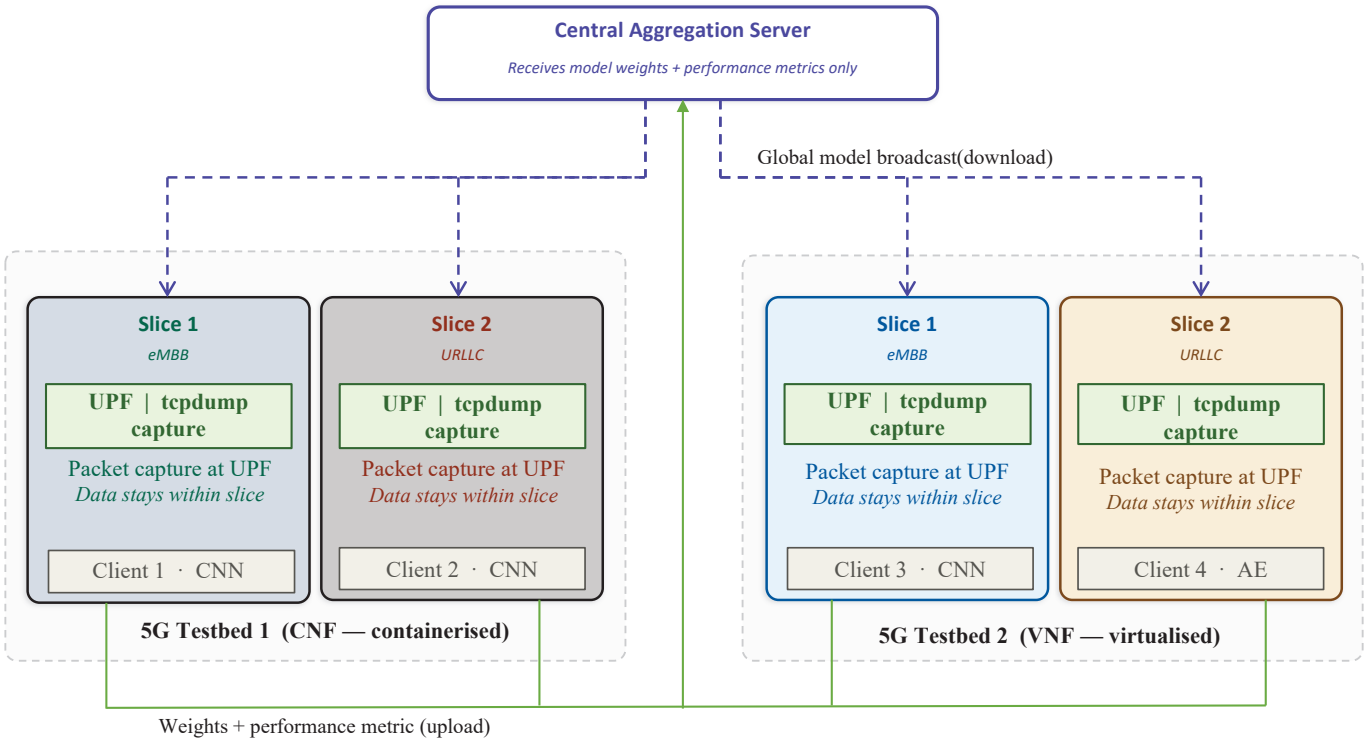


Figure 1 – Federated learning architecture of FLAccShield across heterogeneous 5G testbeds. Each testbed hosts eMBB and URLLC network slices, where traffic is captured at the UPF and used for local model training (CNN or autoencoder). Only model weights and performance metrics are uploaded to the central aggregation server, preserving data privacy within each slice.

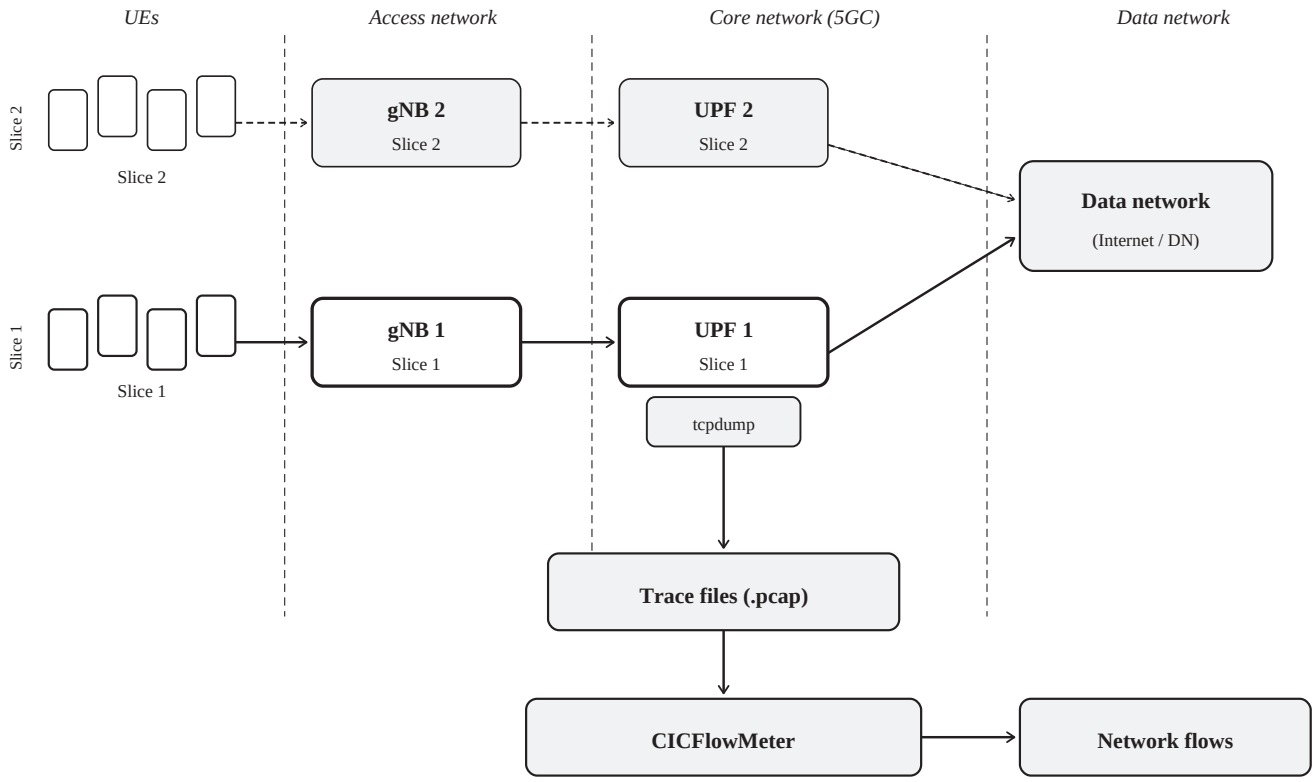


Figure 2 – Traffic generation and processing pipeline [35]

Table 2 – IID scenario (all clients have all 8 attacks)

Attack Types	Client 1	Client 2	Client 3	Client 4
TCP SYN	✓	✓	✓	✓
TCP FIN	✓	✓	✓	✓
TCP ACK	✓	✓	✓	✓
TCP URG	✓	✓	✓	✓
TCP XMAS	✓	✓	✓	✓
TCP YMAS	✓	✓	✓	✓
TCP PUSH	✓	✓	✓	✓
UDP FLOOD	✓	✓	✓	✓

Table 3 – Non-IID scenario (each client has two distinct attacks)

Attack Types	Client 1	Client 2	Client 3	Client 4
TCP SYN			✓	
TCP FIN	✓			
TCP ACK		✓		
TCP URG				✓
TCP XMAS			✓	
TCP YMAS				✓
TCP PUSH		✓		
UDP FLOOD	✓			

In the dataset, each client has over 1 million network flows. For consistency, we have selected 420 000 samples per client with a balanced split of approximately 50% benign and 50% attack traffic. Although benign traffic is a single class, various attack types were stored separately and sampled using stratified sampling to maintain proportional representation. This approach prevented bias toward any specific attack type and preserved the original class distribution. We then split the data for each client into 5 splits: 4 splits were used to create the training set, and the remaining split was used for a test set. With this approach, we created 5-fold datasets, each having a training set and an evaluation set. All of our results are averages from the 5 folds to provide a better estimate of model performance.

Before training, we applied feature scaling using Scikit-learn's `StandardScaler`, which standardizes each feature to have a mean of zero and a standard deviation of one. The scaler was fitted on the training set only and then applied to the test set to prevent data leakage. This step is essential in federated settings to maintain consistency across clients with different data distributions.

The resulting IID and non-IID distributions create distinct client performance profiles during training: under IID conditions, clients converge at comparable rates, while under non-IID conditions, clients with limited attack diversity converge more slowly. This performance asymmetry is the central challenge that FLAccShield's aggregation strategy is designed to address.

3.4 Feature selection and ranking

CICFlowMeter [37] generates 92 flow-based features, resulting in higher computational costs and an increased risk of overfitting. To refine the feature space, we initially applied Principal Component Analysis (PCA) for dimensionality reduction, aiming to retain only the most informative features. However, the model struggled to learn effectively, resulting in suboptimal performance. Another limitation of PCA is that it requires the entire feature set to compute the principal components, meaning that even if a reduced number of components are selected, the transformation still depends on all original features. This dependency introduces additional computational complexity and an additional preprocessing layer, where every sample must first be projected into a new feature space before being used for classification. Given these constraints, we opted for a manual feature evaluation process. We then utilized our domain knowledge to look at every feature in details. We identified features that are solely flow specific (i.e., IP address of source and destination) and do not have any contribution to the model learning. Similarly, we removed a few protocol-specific features, including TCP, UDP, and ICMP-related attributes. Since our dataset encompasses diverse traffic types, features tied to specific protocols did not contribute significantly to a generalized IDS model. Additionally, we identified few redundant features such as total packets, which can be directly inferred from total forward packets and total backward packets. Finally, we discovered few subflow and bulk flow features lacked proper documentation in CICFlowMeter, and their role in network behavior was unclear, making them non-

explainable. After systematically filtering out all the above, we retained 41 unique features. Even with 41 features, the dataset remained extensive, so we ranked them using two distinct ranking approaches: independence score and hierarchical approach.

3.4.1 Feature ranking based on independence score for unsupervised training

For the independence score, similar to [39] we computed the Pearson correlation coefficient scores independently for each slice, allowing us to analyze feature relationships without violating privacy requirements. We have employed an unsupervised feature engineering approach such that the Label column was not considered, and the feature dependencies were evaluated solely based on their relationships with other features. This approach helps us rank features based on their independence score to avoid overfitting. However, the clients showed conflicting outcomes, as a feature ranked highest by one client might be ranked differently by another. Table 4 shows some of the features having multiple ranks across the clients, highlighting the variations in feature importance across different clients.

Table 4 – Feature ranking differences across clients (C)

Flow Feature	Client 1	Client 2	Client 3	Client 4
Bwd Header Length	1	1	3	5
Total Length of Bwd Packet	2	2	2	7
Bwd Packet Length Min	9	12	29	25

To resolve this inconsistency, we averaged the scores across all clients to ensure a unified and unbiased feature ranking. After this step, we established a final ranking from 1 to 41, maintaining a systematic approach to feature selection while adhering to privacy-preserving constraints. In addition to ranking features for our FL framework, we evaluated the performance of individual clients in a local training setting. Unlike the federated approach, where we required a unified ranking, each client in this scenario was independent and ranked its features separately based on its local data.

3.4.2 Feature ranking based on Hierarchical Approach for Supervised Training

Given that our datasets include a Label column, we also apply supervised feature engineering. Based on the Pearson correlation technique, we identified multiple highly correlated features that can be grouped together, with only one feature from each group that must be considered. To implement this, we applied hierarchical clustering across all FL clients, to group similar features together. Fig. 3 provides a visual representation of some of the clusters for client 1 in IID dataset.

Additionally, we calculated the correlation value of each feature with the Label column. In cases where multiple features belonged to the same cluster, we selected the feature with the highest correlation value with the Label column to resolve any ties. After selecting the representative features from each cluster, we ranked the remaining features based on their correlation with the Label column. As a result, each of the four clientss had its own ranked feature set. Due to the removal of some features from clusters, we observed variations in the selected features across different clients. These client-specific features were then used to train models locally in an independent setting. However, since we are targeting an HFL setup, feature sets should be consistent across all clients. When analyzing the clustered features across clients, we found that the clusters were different across clients, and no common clustering pattern could be established to remove the same set of features globally. As a result, performing feature reduction locally was not feasible in the context of federated training. To address this, we opted to retain all 41 features across all clients and instead focused on ranking them globally. Specifically, we computed the Pearson correlation coefficient between each feature and the Label column for every client, then averaged the values across clients to obtain a global correlation score. Based on these scores, the features were ranked in descending order, with higher average correlation values indicating more important features. This unified ranking was then used in the federated training setup to ensure consistency across all clients.

3.5 Design requirements for FLAccShield

The system model and preprocessing pipeline described above establish three concrete requirements for the proposed FL strategy:

1. **Privacy-preserving aggregation:** Raw traffic data must never leave a client's slice boundary; the server interacts only with model weights and scalar performance metrics. It is important to distinguish two senses of privacy here: FLAccShield satisfies *data-residency privacy*, raw traffic never leaves its originating slice privacy, but does not provide *algorithmic privacy guarantees* against inference attacks on shared model weights.
2. **Heterogeneity-aware weighting:** Under non-IID distributions, clients exhibit unequal convergence rates. The aggregation strategy must account for this without requiring additional hyperparameter tuning per deployment.
3. **Dual-paradigm compatibility:** The framework must support both supervised classification (CNN, for attack types present in the training data) and unsupervised anomaly detection (AE, for attack types absent from the training data) under a unified aggregation

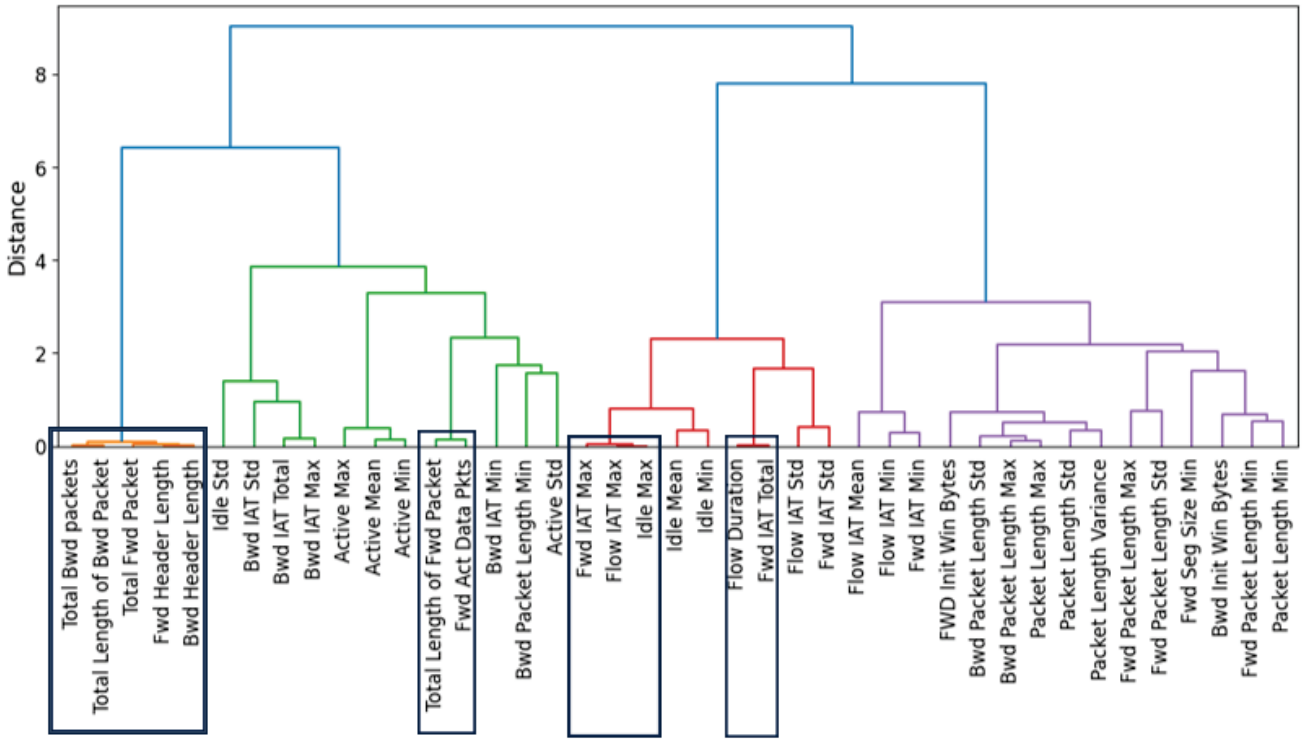


Figure 3 – Hierarchical clustering dendrogram of client 1 for IID data

mechanism, without requiring separate aggregation strategies or additional hyperparameter tuning per paradigm.

Section 4 presents FLAccShield, which is designed to satisfy all three requirements.

4. FLACCSHIELD: PROPOSED FL STRATEGY

In federated learning, client heterogeneity introduces a fundamental tension: clients with diverse data distributions converge at different rates, causing some clients to produce well-optimized local models while others remain underfit. Standard aggregation methods such as FedAvg [24] perform a data-size weighted average of client model weights, treating all updates as equally reliable regardless of local model quality. This uniform treatment amplifies the influence of poorly converged clients, degrading global model performance, particularly under non-IID data distributions common in heterogeneous 5G slice environments.

Existing approaches address this in two opposing ways. FedProx introduces a proximal regularization term $\mu\|w - w^g\|^2$ into each client's local objective, penalizing deviation from the global model to limit the impact of divergent updates. QFedAvg instead assigns higher aggregation weight to clients with larger local loss, accelerating improvement of underperforming participants. Both approaches share two structural limitations: (i) they introduce task-specific hyperparameters (μ for FedProx, q for QFedAvg) that require careful tuning per deployment scenario, and (ii) they are designed around supervised loss signals and do not naturally generalize to unsupervised anomaly detection settings, where no label-derived loss is available.

We propose Federated Accuracy Shield (FLAccShield), an aggregation strategy that addresses client heterogeneity by weighting model contributions according to each client's local validation performance, specifically, validation accuracy for supervised settings and reconstruction error for unsupervised settings. Critically, this performance signal is computed on each client's local held-out data and communicated to the server alongside model weights, requiring no modification to local loss functions and introducing no additional hyperparameters beyond those already present in standard FL training. This design makes FLAccShield inherently compatible with both learning paradigms without retuning.

4.1 Performance-weighted aggregation

The core insight of FLAccShield is that a client's local validation performance provides a reliable proxy for the quality of its model update. A client with a high validation accuracy has converged toward a well-fitting local optimum; its update should exert strong influence on the global model. Conversely, a client with low accu-

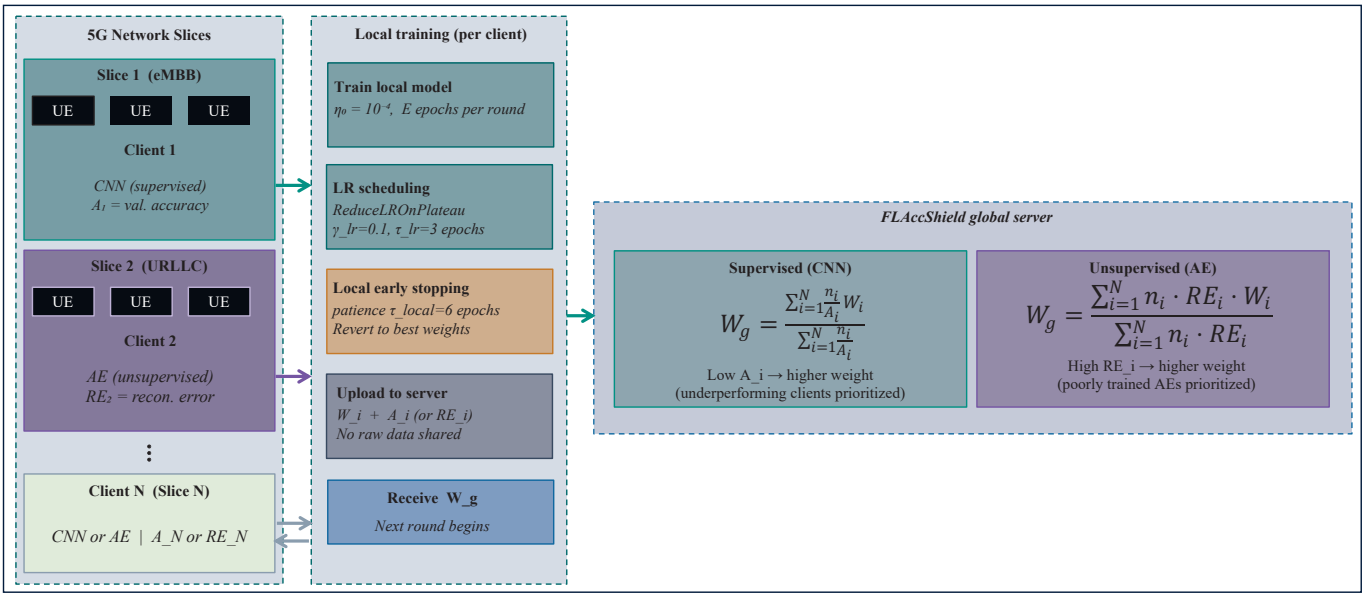


Figure 4 – Overview of the proposed FLAccShield framework for federated learning across 5G network slices, where clients train local CNN/AE models and upload model weights with performance metrics for adaptive global aggregation.

racy has not yet converged and its update, if weighted equally, would pull the global model away from better solutions. FLAccShield encodes this intuition by weighting each client's contribution inversely proportional to its validation accuracy.

4.1.1 Supervised aggregation

Let W_i denote the model weights submitted by client i, $A_i \in (0, 1]$ its local validation accuracy, and n_i the number of training samples it holds. The global model weights W_g are computed as:

$$W_g = \frac{\sum_{i=1}^{N} \frac{n_i}{A_i} W_i}{\sum_{i=1}^{N} \frac{n_i}{A_i}} \qquad (1)$$

The weight assigned to client i is $\frac{n_i}{A_i}$: the data-size factor n_i follows the FedAvg convention of proportional representation, while the $\frac{1}{A_i}$ factor amplifies the contribution of low-accuracy clients. This prioritizes underperforming clients in the global aggregation, allowing high-performing clients to refine their updates in subsequent rounds and progressively improve the weakest participants through collaborative knowledge transfer. Note that when all clients achieve identical accuracy, Eq. *(1)* reduces to the standard FedAvg weighted average, confirming that FLAccShield subsumes FedAvg as a special case.

4.1.2 Unsupervised aggregation

For anomaly detection, client models are Autoencoders (AEs) trained exclusively on benign traffic. No class labels are available, so validation accuracy cannot be computed. Instead, we use the mean reconstruction error RE_i on the client's local held-out benign set as the performance signal. A higher reconstruction error indicates a less well-trained encoder, so client contributions are weighted *proportionally* to their reconstruction error, the inverse of the supervised case:

$$W_g = \frac{\sum_{i=1}^{N} n_i \cdot RE_i \cdot W_i}{\sum_{i=1}^{N} n_i \cdot RE_i} \qquad (2)$$

This gives greater influence to clients whose autoencoders reconstruct benign traffic poorly, precisely those whose local training has benefited least and whose updates carry the most unexploited information for the global model to absorb.

The notation shared by both equations is as follows: W_g — aggregated global model weights; W_i — local model weights of client i; A_i — local validation accuracy of client i; RE_i — mean reconstruction error of client i on held-out benign data; n_i — number of training samples at client i; N — total number of participating clients.

4.2 Convergence efficiency mechanisms

Beyond aggregation, FLAccShield incorporates two mechanisms to improve training stability and reduce unnecessary computation.

4.2.1 Adaptive learning rate scheduling

Each client begins every round with an initial learning rate of $\eta_0 = 10^{-4}$. A `ReduceLROnPlateau` scheduler monitors local validation accuracy and reduces the learning rate by a factor of $\gamma_{lr} = 0.1$ when no improvement is observed for $\tau_{lr} = 3$ consecutive epochs. The minimum permissible learning rate is $\eta_{min} = 10^{-6}$, allowing at most three reductions per round given the epoch budget. This prevents premature convergence to poor local minima while avoiding oscillation in later training stages.

4.2.2 Two-level early stopping

FLAccShield applies early stopping at both the local and global levels to prevent overfitting and eliminate redundant computation.

Local early stopping: Each client monitors validation accuracy per epoch. If no improvement is observed for $\tau_{local} = 6$ consecutive epochs, local training is terminated and the best recorded weights are transmitted to the server. The patience $\tau_{local} = 2\tau_{lr}$ is deliberately set to twice the scheduler patience, ensuring the model has sufficient time to adapt following a learning rate reduction before stopping is enforced.

Global early stopping: After each aggregation round, the server evaluates whether the updated global model represents genuine improvement. Specifically, if fewer than 50% of clients achieve a validation accuracy gain exceeding $\delta = 0.01$ over the previous round's global model, training is halted. This criterion detects the onset of global overfitting, a state where additional rounds produce diminishing or negative returns, and prevents unnecessary communication overhead in later rounds.

4.3 Model architectures

FLAccShield is evaluated with two deep learning architectures, each suited to its respective learning paradigm.

4.3.1 CNN for supervised DDoS classification

For known-attack detection, we adopt the 1D-CNN architecture proposed in [11], which treats the ordered feature vector as a 1D signal and applies convolutional filters to extract local feature correlations indicative of DDoS behavior. The architecture, detailed in Table 5, comprises two convolutional blocks followed by max-pooling for dimensionality reduction, and two fully connected layers for binary classification. The output layer uses a sigmoid activation to produce a probability over the two classes (benign, attack). The model accepts variable-length input, accommodating feature counts ranging from 4 to 41.

Table 5 – CNN architecture (supervised training)

Layer	Shape	Details	Activation
Input	$(N, F, 1)$	Features as 1D	–
Conv1D	$(N, F-2, 6)$	6 filters, 3×1	ReLU
MaxPool1D	$(N, \frac{F-2}{2}, 6)$	Pool 2×1	–
Conv1D	$(N, \frac{F-4}{2}, 16)$	16 filters, 3×1	ReLU
MaxPool1D	$(N, \frac{F-4}{4}, 16)$	Pool 2×1	–
Flatten	–	To vector	–
Dense	120	Fully connected	ReLU
Dense	84	Fully connected	ReLU
Output	2 (binary)	Classification	Sigmoid

4.3.2 Autoencoder for unsupervised anomaly detection

For detecting attack categories absent from the training distribution, we employ a symmetric autoencoder trained exclusively on benign traffic. The encoder compresses the input feature vector through two hidden layers to a latent representation of dimension L; the decoder symmetrically reconstructs the original input. The architecture is described in Table 6. During inference, a flow is flagged as anomalous if its reconstruction error exceeds a fixed threshold $\theta = 0.2$. This threshold was selected empirically: percentile-based thresholds (e.g., 95th percentile of training reconstruction errors) and dynamic best-F1 selection were evaluated but produced inconsistent behavior across feature counts and folds. A fixed threshold ensures a fair, reproducible comparison across all strategies and feature set sizes.

4.4 Training pipeline

Algorithm 1 formalizes the complete FLAccShield training pipeline. The outer loop iterates over global rounds; within each round, all N clients independently execute local training subject to early stopping and learning rate scheduling, then transmit their model weights and performance signal to the server. The server performs performance-weighted aggregation (Eq. *(1)* or Eq. *(2)* depending on mode $\mathcal{M}$), broadcasts the updated global weights, and evaluates the global early stopping criterion.

Table 6 – Autoencoder architecture

Layer	Output Shape	Details	Activation
Input	(N, F)	Features (4–41)	–
Dense (Enc)	–	Hidden layer 1 (compression)	ReLU
Dense (Enc)	–	Hidden layer 2 (deeper encoding)	ReLU
Latent	(N, L)	Bottleneck feature representation	ReLU
Dense (Dec)	–	Expand latent vector	ReLU
Dense (Dec)	–	Approximate original dimensions	ReLU
Output	(N, F)	Reconstructed input	Sigmoid

This cycle repeats until the stopping condition is met or the maximum round budget R is exhausted.

FLAccShield is implemented using the Flower framework [40], which provides the client-server communication layer. Both CNN and AE models accept variable input dimensions, allowing the same codebase to evaluate all feature set sizes without architectural changes. The full experiment suite, spanning feature counts, folds, and FL strategies, is automated via a configuration-driven bash script that parameterizes each training run.

The components of FLAccShield operate at two distinct levels of generality. The core aggregation strategy, performance-weighted aggregation using inverse validation accuracy for supervised settings and reconstruction error weighting for unsupervised settings, is a general-purpose FL mechanism that requires no knowledge of the underlying task domain, introduces no domain-specific hyperparameters, and is broadly applicable to federated learning scenarios where clients produce heterogeneous local models, as demonstrated here under both IID and non-IID data distributions. Similarly, the two-level early stopping mechanism and adaptive learning rate scheduling are task-agnostic convergence controls applicable to any FL training pipeline. In contrast, the instantiation presented in this paper makes several design choices specific to DDoS detection in 5G network slices: the 1D-CNN and fully-connected autoencoder architectures reflect the structure of network traffic data; the data collection pipeline—PCAP capture at the UPF using `tcpdump`, conversion to flow-based features via CICFlowMeter, and the 41-feature space, is specific to the network intrusion detection domain; and the fixed reconstruction error threshold $\theta = 0.2$ was selected empirically for this dataset and would require re-calibration in other application domains. Researchers applying FLAccShield to other federated learning settings would retain the aggregation equations and convergence controls while substituting task-appropriate model architectures and preprocessing pipelines.

Algorithm 1 FL Training Pipeline with FLAccShield

Input: Feature set F, Fold set K, Clients C, Rounds R, Mode $\mathcal{M} \in \{\text{Supervised}, \text{Unsupervised}\}$

Hyperparameters:
τ_{local}: Local Early Stopping patience (epochs)
τ_{lr}: Learning Rate Scheduler patience (epochs)
γ_{lr}: Learning rate reduction factor
τ_{global}: Global Early Stopping patience (rounds)
Repeat for all features $f \in F$ **and folds** $k \in K$**:**

Initialize Global Server $S_{f,k}$ and Clients $C_{f,k}$
for each global round $r \in \{1, ..., R\}$ **do**
  for each client $c \in C_{f,k}$ **do**
    for each epoch e **do**
      Train local model for one epoch
      if no validation improvement for τ_{local} epochs **then**
        Apply Local Early Stopping
      end if
      if no improvement for τ_{lr} epochs **then**
        Reduce learning rate by γ_{lr}
      end if
    end for
    if $\mathcal{M}$ = Supervised **then**
      Send model weights and validation accuracy to $S_{f,k}$
    else
      Send model weights and reconstruction error to $S_{f,k}$
    end if
  end for
  Global Aggregation:
  if $\mathcal{M}$ = Supervised **then**
    Aggregate using inverse of validation accuracy (Eq. *(1)*)
  else
    Aggregate using reconstruction error (Eq. *(2)*)
  end if
  if no global performance improvement for τ_{global} rounds **then**
    Apply Global Early Stopping
  end if
end for
Save best global model $W_{f,k}$ as '.pth' file

4.5 Hyperparameter summary

Table 7 consolidates all hyperparameters used in the FLAccShield training pipeline along with their values and justifications.

Table 7 – FLAccShield hyperparameters

Hyperparameter	Value	Description
Initial LR η_0	10^{-4}	Standard Adam rate
LR reduction γ_{lr}	0.1	Conservative steps
LR patience τ_{lr}	3 epochs	Local plateau response
Min LR η_{min}	10^{-6}	Prevents degeneration
Local ES patience	6 epochs	$2 \times \tau_{lr}$ adaptation
Global ES threshold δ	0.01	Min accuracy gain
Global ES clients	50%	Majority stopping
Max rounds R	20	Communication limit
Max local epochs	30	Computation limit
Error threshold θ	0.2	Fixed reconstruction
Client participation	All (N)	Full stability

5. EXPERIMENT AND RESULT ANALYSIS

We evaluate FLAccShield against three baseline FL strategies, FedAvg, FedProx, and QFedAvg, as well as a non-federated independent local training baseline, where each client trains exclusively on its own data. Experiments are conducted across three settings summarized in Table 8: supervised classification under IID and non-IID data distributions, and unsupervised anomaly detection under IID conditions. Each setting is evaluated over a range of feature set sizes to assess the joint impact of feature engineering and model aggregation strategy on detection performance. All results are averaged over 5-fold cross-validation, with F1 score as the primary evaluation metric given its balanced treatment of precision and recall across varying data distributions.

In the federated setting, performance is tracked at two levels per round: the local model updated by each client, and the global model returned by the server. For supervised training, both are evaluated using validation accuracy; for unsupervised training, reconstruction error on held-out benign traffic is used. In the local training baseline, models are evaluated after each epoch using the same respective metrics. This dual-level tracking allows us to assess both individual client convergence and the effectiveness of global aggregation across rounds.

5.1 FLAccShield classification performance analysis

5.1.1 Experiment 1.1: Supervised classification with IID dataset

Fig. 6 shows the average F1 score comparison among different strategies, as well as independent local training. Although all the strategies have an F1 score of 0.99 for feature count 4, we observe some fluctuations in performance for all baseline strategies as the feature number increases. Since we used supervised feature ranking, adding more features encourages overfitting and forces some clients to deviate in their performance. As our proposed strategy enforces low-performing clients to follow high-performing clients, it addresses this overfitting issue and maintains superior performance across varied feature sets.

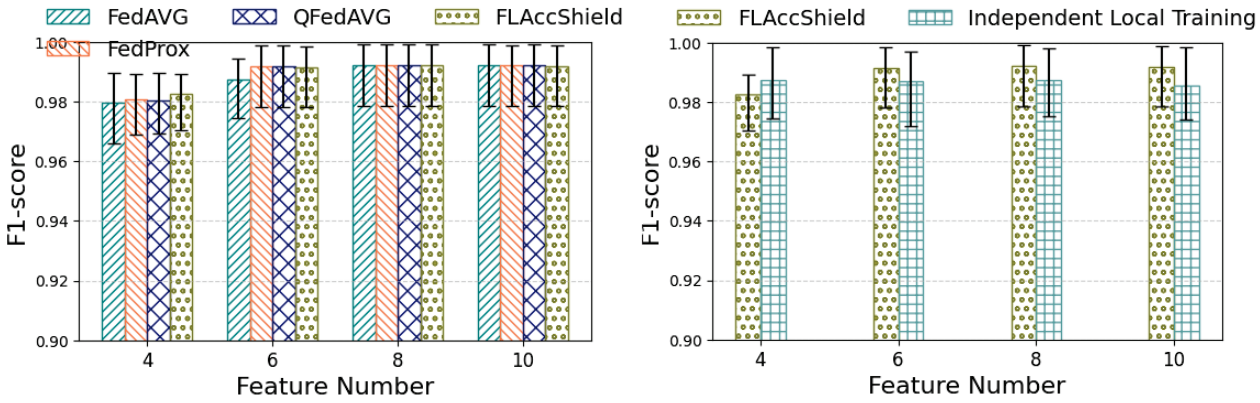
Table 9 demonstrates client-wise F1 scores, and it is evident that FLAccShield helped all clients successfully defend against this deviation and maintain uniform performance. Moreover, FLAccShield outperforms local training for different counts of features. Unlike local training, which restricts each model to learning only from its own data, FL allows each client to benefit from the collective knowledge of the other clients.

This phenomenon enables the clients to learn from a broader range of data representations. In addition, FLAccShield enhances the performance of low-performing clients by prioritizing their contributions during model aggregation, which is a benefit not available in local training.

5.1.2 Experiment 1.2: Supervised training with non-IID dataset

In this experiment, we evaluated the effectiveness of our proposed strategy, as well as other baseline strategies on a non-IID version of the dataset. We have already discussed the traffic distribution of non-IID datasets, as well as the training and test set samples in previous sections.

Fig. 7 presents the average F1 scores for all strategies, including independent local training. For feature number 4, FLAccShield, QFedAVG, FedProx, and FedAVG have F1 scores of 0.98, 0.97, 0.97, and 0.91, respectively. Similar to Experiment 1.1, adding more features is likely to introduce overfitting, and all strategies start to degrade in performance. FLAccShield showed an 8% decrease in F1 score between feature numbers 4 and 10. At the same step, QFedAVG, FedProx, and FedAVG showed deviations of 22%, 15%, and 12%, respectively.



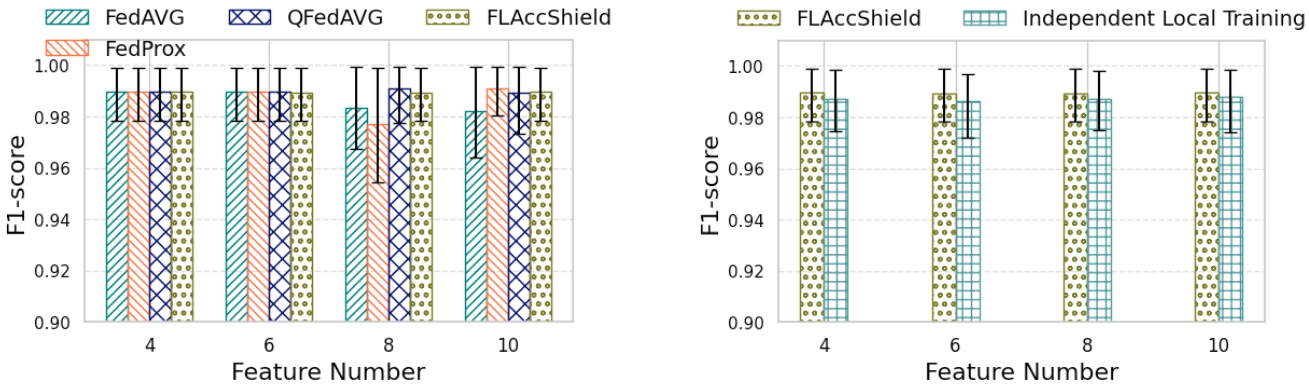
(a) FL Strategies

(b) FL vs local training

Figure 5 – Supervised training with IID dataset with 6 clients: Average F1 score comparison

Table 8 – Overview of conducted experiments

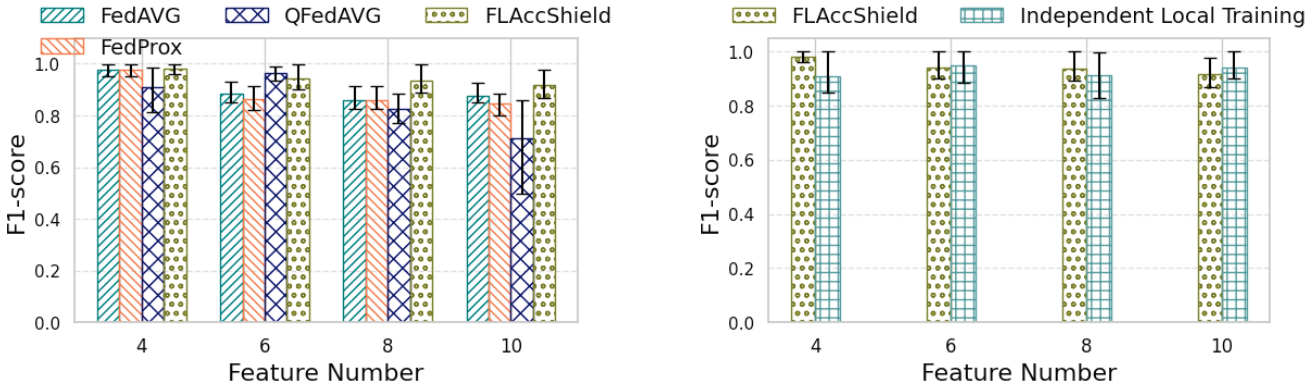
#	Training	Experiment	Feature Ranking	Data Dist.
1	Supervised	1.1	Hierarchical Cluster	IID
		1.2	Hierarchical Cluster	non-IID
2	Unsupervised	2	independence score	IID



(a) FL Strategies

(b) FL vs local training

Figure 6 – Supervised training with IID Dataset: Average F1 score comparison



(a) FL Strategies

(b) FL vs local training

Figure 7 – Supervised training with non-IID dataset: Average F1 score comparison

Table 9 – Per-client F1 scores (mean ± SD) for IID, supervised CNN.

Strategy	Feature count	Client 1	Client 2	Client 3	Client 4
FLAccShield	4	0.9785 ± 0.0007	0.9817 ± 0.0006	0.9990 ± 0.0001	0.9991 ± 0.0002
	6	0.9783 ± 0.0008	0.9816 ± 0.0007	0.9989 ± 0.0002	0.9989 ± 0.0001
	8	0.9782 ± 0.0008	0.9816 ± 0.0005	0.9988 ± 0.0003	0.9988 ± 0.0004
	10	0.9784 ± 0.0008	0.9817 ± 0.0006	0.9990 ± 0.0001	0.9990 ± 0.0002
FedAvg	4	0.9785 ± 0.0007	0.9817 ± 0.0006	0.9990 ± 0.0001	0.9991 ± 0.0002
	6	0.9785 ± 0.0008	0.9817 ± 0.0006	0.9990 ± 0.0002	0.9990 ± 0.0002
	8	0.9672 ± 0.0273	0.9676 ± 0.0269	0.9993 ± 0.0001	0.9992 ± 0.0002
	10	0.9642 ± 0.0230	0.9652 ± 0.0238	0.9993 ± 0.0002	0.9993 ± 0.0002
FedProx	4	0.9785 ± 0.0007	0.9818 ± 0.0006	0.9990 ± 0.0001	0.9991 ± 0.0002
	6	0.9784 ± 0.0008	0.9818 ± 0.0006	0.9990 ± 0.0001	0.9990 ± 0.0003
	8	0.9546 ± 0.0243	0.9551 ± 0.0243	0.9991 ± 0.0001	0.9991 ± 0.0002
	10	0.9803 ± 0.0210	0.9846 ± 0.0224	0.9992 ± 0.0001	0.9992 ± 0.0003
QFedAvg	4	0.9784 ± 0.0008	0.9817 ± 0.0006	0.9990 ± 0.0002	0.9990 ± 0.0001
	6	0.9785 ± 0.0007	0.9818 ± 0.0006	0.9990 ± 0.0001	0.9991 ± 0.0002
	8	0.9871 ± 0.0216	0.9776 ± 0.0267	0.9992 ± 0.0002	0.9992 ± 0.0002
	10	0.9734 ± 0.0241	0.9847 ± 0.0214	0.9992 ± 0.0003	0.9992 ± 0.0002
Local training	4	0.9744 ± 0.0063	0.9789 ± 0.0040	0.9984 ± 0.0001	0.9964 ± 0.0039
	6	0.9719 ± 0.0030	0.9797 ± 0.0040	0.9970 ± 0.0037	0.9968 ± 0.0042
	8	0.9751 ± 0.0045	0.9778 ± 0.0046	0.9982 ± 0.0010	0.9974 ± 0.0028
	10	0.9742 ± 0.0055	0.9815 ± 0.0008	0.9971 ± 0.0034	0.9986 ± 0.0006

Table 7 shows F1 scores for individual client where each client in FLAccShield is showing a stable F1 scores compared to others which is a proof to address overfitting by FLAccShield that helps to outperform other strategies for both (lower or higher) number of features. Unlike FLAccShield, local training shows less overfitting as the feature number increases. In fact, it helps improve the performance in some cases. In local training, each client has access to a limited number of attack types, and the addition of more features might help the model learn more patterns from the dataset. On the other hand, during FL training, as clients have already gained knowledge of unknown attacks from other clients through collaboration, adding additional features may slightly cause overfitting.

5.1.3 *Experiment 2: Unsupervised training with IID dataset*

In this experiment, we evaluate FLAccShield's ability to detect attack categories entirely absent from the training data, a capability we term unknown attack detection. As we already stated in Section 3.3, benign traffic does not have segregation into different sub-classes, and there is no variation in benign traffic between IID and non-IID datasets, which makes it irrelevant to perform anomaly detection with the non-IID dataset. Although the trained set has been reduced to half from supervised samples due to considering only benign traffic, the test set remains 84 000 with a 1:1 ratio of benign traffic and attacks. We have trained the model with a varied range of features where feature ranking was calculated with an unsupervised technique, and we detail the process in Section 3.4.1. To classify anomalies as well as to calculate the F1 score, we set 0.2 as a reconstruction error threshold, a hyperparameter described in the hyperparameter section.

Fig. 8 shows the average F1 score across all clients for different feature counts, while Table 11 provides client-wise F1 scores. Compared to the supervised approach, a significantly larger number of features was needed to achieve a 0.9 F1 score. This is understandable, as we did not directly consider the relationship with the label column, requiring more features to reach a high F1 score. From the Fig., we can clearly observe that all of the strategies, including the proposed and baseline

Table 10 – Per-client F1 scores (mean ± SD) for non-IID, supervised CNN.

Strategy	Feature count	Client 1	Client 2	Client 3	Client 4
FLAccShield	4	0.9589 ± 0.0168	0.9589 ± 0.0169	0.9995 ± 0.0006	0.9993 ± 0.0016
	6	0.9012 ± 0.0865	0.9233 ± 0.0941	0.9990 ± 0.0013	0.9453 ± 0.0497
	8	0.9033 ± 0.0663	0.8905 ± 0.0622	0.9991 ± 0.0012	0.9468 ± 0.0483
	10	0.8693 ± 0.0514	0.8918 ± 0.0758	0.9752 ± 0.0549	0.9275 ± 0.0405
FedAvg	4	0.9513 ± 0.0004	0.9501 ± 0.0028	0.9997 ± 0.0002	0.9999 ± 0.0001
	6	0.8496 ± 0.0570	0.8505 ± 0.0563	0.9014 ± 0.0551	0.9295 ± 0.0394
	8	0.8235 ± 0.0033	0.8235 ± 0.0032	0.8768 ± 0.0004	0.9118 ± 0.0007
	10	0.8507 ± 0.0562	0.8495 ± 0.0571	0.8768 ± 0.0004	0.9280 ± 0.0404
FedProx	4	0.9514 ± 0.0004	0.9500 ± 0.0032	0.9995 ± 0.0008	0.9987 ± 0.0028
	6	0.8235 ± 0.0032	0.8222 ± 0.0028	0.9014 ± 0.0551	0.9118 ± 0.0007
	8	0.8235 ± 0.0034	0.8248 ± 0.0030	0.8768 ± 0.0004	0.9118 ± 0.0007
	10	0.8005 ± 0.0367	0.8260 ± 0.0025	0.8749 ± 0.0044	0.8860 ± 0.0348
QFedAvg	4	0.8127 ± 0.1021	0.9706 ± 0.0242	0.9848 ± 0.0047	0.8705 ± 0.0644
	6	0.9340 ± 0.1031	0.9901 ± 0.0022	0.9888 ± 0.0070	0.9451 ± 0.0636
	8	0.7964 ± 0.0366	0.7716 ± 0.0675	0.8465 ± 0.0411	0.8862 ± 0.0353
	10	0.7695 ± 0.0289	0.7209 ± 0.0558	0.4955 ± 0.4533	0.8605 ± 0.0286
Local training	4	0.8942 ± 0.1287	0.8919 ± 0.0916	0.9995 ± 0.0000	0.8475 ± 0.0007
	6	0.9430 ± 0.1062	0.9587 ± 0.0731	0.9995 ± 0.0000	0.8853 ± 0.0353
	8	0.8293 ± 0.0691	0.9170 ± 0.0750	0.9986 ± 0.0014	0.8972 ± 0.0279
	10	0.9020 ± 0.0501	0.9567 ± 0.0491	0.9988 ± 0.0021	0.9110 ± 0.0007

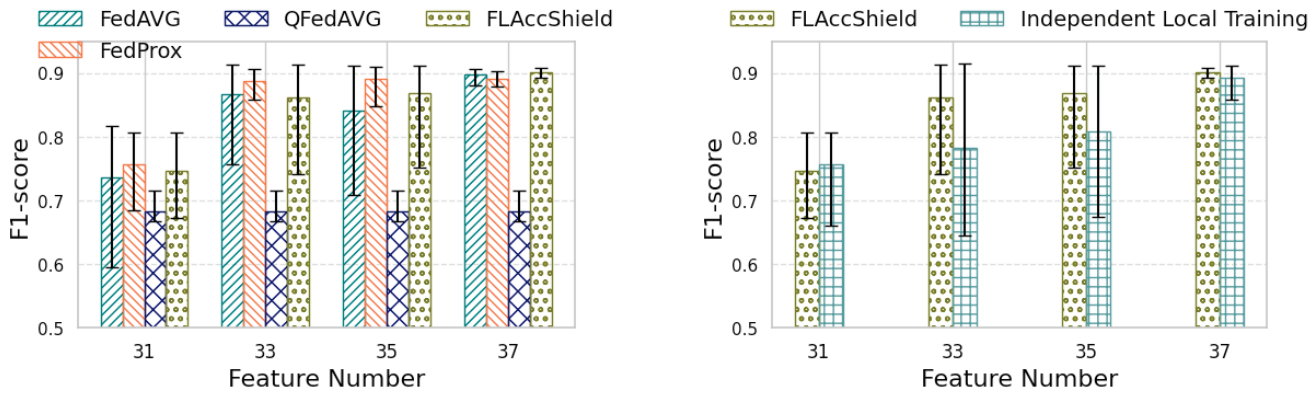
ones, improve performance as the number of features increases.

With a lower number of features, i.e., 31, 33, the performance of FLAccShield is slightly lower than FedAVG and FedProx, but as the feature count grows, FLAccShield improves more sharply than others and at a point (feature 37) it outperforms others: QFedAvg by 32%, FedProx, independent local training by 1%, and FedAvg by 0.4%. Another significant observation here is that QFedAVG struggles a lot to detect anomalies compared to its supervised performances (experiments 1.1 and 1.2).

In the FL training global model learns the knowledge from all of the clients through collaboration. Our setup has four clients, meaning FL model has access to 3 times more knowledge and variation compared to local training happening silo basis which is helping FL to outperform independent local training.

5.2 Inference time analysis

While evaluating performance across different feature counts, we made an important observation regarding computational complexity. Initially, we tried to analyze training time in relation to the number of features. However, since the training process used GPU acceleration, the impact of feature count was not clearly visible. GPUs handle parallel computations efficiently, which makes the training time appear mostly stable regardless of feature dimensionality. To better understand the computational burden, we focused on inference time instead. Inference is generally less resource-intensive and is often performed on devices without GPU support [41]. To simulate a realistic scenario, we measured inference time using only a CPU for all FL strategies and the local training approach. The inference time measurements were taken from Experiment 2, which involved AE-based anomaly detection using unsupervised learning. As shown in Fig. 9, inference time increases significantly with a higher number of features. Specifically, when the number of features reaches 38 or more, the inference time nearly doubles compared to using a lower number of features. For both supervised experiments (Ex 1.1 and 1.2), inference time



(a) FL Strategies

(b) FL vs local training

Figure 8 – Unsupervised model with IID dataset: Average F1 score comparison

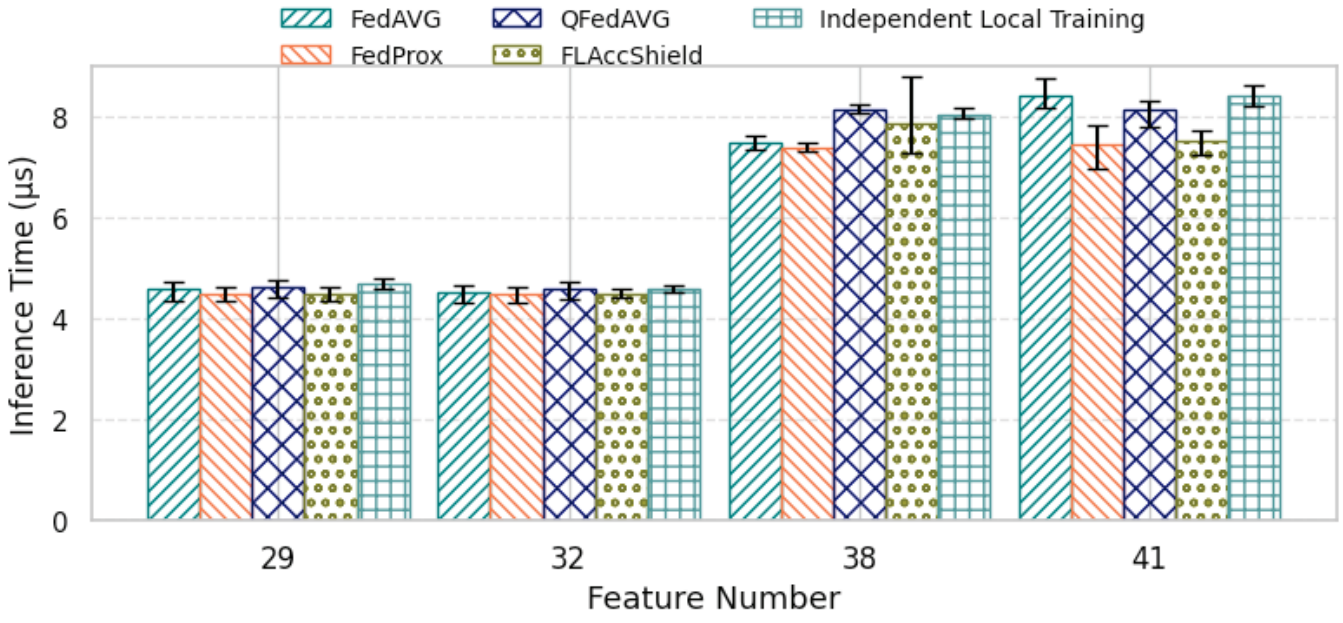


Figure 9 – Inference time per sample (unsupervised model with IID dataset)

Table 11 – Per-client F1 scores (mean ± SD) for IID, unsupervised AE.

Strategy	Feature count	Client 1	Client 2	Client 3	Client 4
FLAccShield	31	0.73176 ± 0.00081	0.67223 ± 0.00068	0.80622 ± 0.00053	0.77891 ± 0.08440
	33	0.88181 ± 0.02715	0.74193 ± 0.06910	0.91412 ± 0.00070	0.90794 ± 0.00498
	35	0.90112 ± 0.02217	0.75235 ± 0.07285	0.91176 ± 0.00108	0.90636 ± 0.00361
	37	0.90607 ± 0.00089	0.89330 ± 0.02479	0.90758 ± 0.00138	0.89959 ± 0.00551
FedAvg	31	0.72868 ± 0.00741	0.59564 ± 0.17196	0.80603 ± 0.00071	0.81713 ± 0.00360
	33	0.89204 ± 0.02772	0.75610 ± 0.05812	0.91339 ± 0.00111	0.90722 ± 0.00390
	35	0.83947 ± 0.10909	0.70883 ± 0.12785	0.91162 ± 0.00099	0.90631 ± 0.00428
	37	0.90643 ± 0.00109	0.88096 ± 0.02364	0.90680 ± 0.00055	0.89823 ± 0.00401
FedProx	31	0.74141 ± 0.02788	0.68434 ± 0.03090	0.79848 ± 0.00237	0.80588 ± 0.00548
	33	0.88617 ± 0.05247	0.85883 ± 0.04940	0.90646 ± 0.00233	0.89711 ± 0.00421
	35	0.91023 ± 0.00081	0.84865 ± 0.03108	0.90524 ± 0.00275	0.89755 ± 0.00278
	37	0.90325 ± 0.00152	0.87886 ± 0.02314	0.89688 ± 0.00396	0.88344 ± 0.00499
QFedAvg	31	0.68108 ± 0.00001	0.71553 ± 0.00001	0.66697 ± 0.00000	0.66715 ± 0.00001
	33	0.68034 ± 0.00001	0.71628 ± 0.00001	0.66757 ± 0.00000	0.66653 ± 0.00001
	35	0.68287 ± 0.00001	0.71537 ± 0.00001	0.66721 ± 0.00000	0.66528 ± 0.00001
	37	0.68285 ± 0.00001	0.71394 ± 0.00001	0.66653 ± 0.00000	0.66741 ± 0.00001
Local training	31	0.78080 ± 0.00055	0.66030 ± 0.01429	0.80663 ± 0.00094	0.77996 ± 0.08585
	33	0.75436 ± 0.03408	0.64535 ± 0.00073	0.91483 ± 0.00092	0.81577 ± 0.00326
	35	0.74094 ± 0.03387	0.67445 ± 0.03069	0.91197 ± 0.00103	0.90563 ± 0.00372
	37	0.89590 ± 0.00099	0.85771 ± 0.00037	0.91181 ± 0.00082	0.90426 ± 0.00367

remains the same due to the low feature counts and is therefore omitted.

Since all models used the same AE architecture, the increase in inference time is consistent across all strategies. This clearly shows that higher-dimensional inputs introduce noticeable computational costs during inference. In FL environments, devices with different hardware capabilities often participate in inference. Some may have limited processing power, especially those relying on smaller CPUs. In such cases, keeping the feature count lower can improve overall system efficiency. Reducing the number of features not only helps minimize computational load but also supports smoother deployment across a wide range of devices.

5.3 Impact of increased number of clients

In our prior experiments presented in Table 8, the FL setup involved four clientss, each of which represents a network slice. To evaluate the scalability and robustness of the proposed FLAccShield strategy, we extended our setup by introducing two additional clients, derived from two new network slices hosted in the containerized testbed.

Each newly added slice, consistent with the original configuration, hosts 4 UEs. We followed the same traffic generation methodology described in Section 3.2, including automated benign traffic generation through web browsing, video streaming, file transfers, and attack traffic generation using the hping3 tool to simulate the 8 DDoS attack types listed in Table 2.

The traffic was preprocessed, standardized, and converted into flow-based IID datasets using the same pipeline detailed in Fig. 2. Feature selection followed the hierarchical ranking approach described in Section 3.4. Each client dataset was fixed at 420 000 samples, maintaining a balanced ratio of benign and attack traffic, and split into training and test sets following the IID distribution. The primary goal of this experiment is to replicate Experiment 1.1 under the expanded 6-client setting, and analyze how the increased number of clients influences model performance.

Fig. 5 presents the average F1 score comparison between FLAccShield, baseline FL strategies, and independent local training, evaluated using hierarchical feature sets. All FL strategies, including FLAccShield, demonstrated strong performance (F1 score - 0.98) with only 4 features. As the number of features increased, the F1 scores initially

Table 12 – Per-client F1 scores (mean ± SD) for six-client IID, supervised CNN.

Strategy	Feature count	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6
FLAccShield	4	0.97036 ± 0.00125	0.97473 ± 0.00044	0.98611 ± 0.00448	0.98563 ± 0.00443	0.98935 ± 0.00045	0.98823 ± 0.00081
	6	0.97797 ± 0.00071	0.98145 ± 0.00079	0.99844 ± 0.00035	0.99856 ± 0.00027	0.99727 ± 0.00053	0.99586 ± 0.00037
	8	0.97843 ± 0.00078	0.98175 ± 0.00057	0.99899 ± 0.00010	0.99903 ± 0.00018	0.99784 ± 0.00068	0.99644 ± 0.00050
	10	0.97841 ± 0.00081	0.98167 ± 0.00054	0.99891 ± 0.00018	0.99892 ± 0.00017	0.99757 ± 0.00053	0.99624 ± 0.00056
FedAvg	4	0.96927 ± 0.00131	0.96610 ± 0.01877	0.98269 ± 0.00067	0.98217 ± 0.00107	0.98952 ± 0.00031	0.98840 ± 0.00062
	6	0.97452 ± 0.00520	0.97852 ± 0.00393	0.99229 ± 0.00898	0.99201 ± 0.00921	0.99431 ± 0.00440	0.99280 ± 0.00443
	8	0.97853 ± 0.00075	0.98175 ± 0.00057	0.99905 ± 0.00012	0.99907 ± 0.00018	0.99802 ± 0.00027	0.99668 ± 0.00020
	10	0.97853 ± 0.00074	0.98174 ± 0.00057	0.99905 ± 0.00012	0.99907 ± 0.00018	0.99802 ± 0.00027	0.99665 ± 0.00021
FedProx	4	0.96896 ± 0.00112	0.97423 ± 0.00071	0.98245 ± 0.00125	0.98200 ± 0.00094	0.98934 ± 0.00061	0.98818 ± 0.00048
	6	0.97822 ± 0.00096	0.98165 ± 0.00076	0.99883 ± 0.00021	0.99887 ± 0.00031	0.99745 ± 0.00068	0.99604 ± 0.00038
	8	0.97853 ± 0.00074	0.98175 ± 0.00057	0.99905 ± 0.00012	0.99907 ± 0.00018	0.99784 ± 0.00067	0.99645 ± 0.00050
	10	0.97853 ± 0.00074	0.98175 ± 0.00057	0.99887 ± 0.00017	0.99895 ± 0.00024	0.99800 ± 0.00026	0.99664 ± 0.00022
QFedAvg	4	0.96909 ± 0.00112	0.97405 ± 0.00078	0.98103 ± 0.00260	0.98033 ± 0.00240	0.98953 ± 0.00030	0.98839 ± 0.00063
	6	0.97831 ± 0.00083	0.98157 ± 0.00060	0.99892 ± 0.00022	0.99893 ± 0.00008	0.99744 ± 0.00066	0.99608 ± 0.00071
	8	0.97853 ± 0.00074	0.98175 ± 0.00058	0.99905 ± 0.00012	0.99907 ± 0.00018	0.99785 ± 0.00038	0.99650 ± 0.00054
	10	0.97853 ± 0.00074	0.98175 ± 0.00057	0.99901 ± 0.00015	0.99903 ± 0.00025	0.99783 ± 0.00044	0.99643 ± 0.00038
Local training	4	0.97437 ± 0.00626	0.97891 ± 0.00402	0.99838 ± 0.00011	0.99640 ± 0.00394	0.98782 ± 0.00136	0.98714 ± 0.00112
	6	0.97193 ± 0.00303	0.97968 ± 0.00397	0.99697 ± 0.00372	0.99677 ± 0.00416	0.98844 ± 0.00035	0.98728 ± 0.00071
	8	0.97505 ± 0.00448	0.97780 ± 0.00462	0.99823 ± 0.00096	0.99735 ± 0.00282	0.98837 ± 0.00058	0.98746 ± 0.00075
	10	0.97420 ± 0.00553	0.98147 ± 0.00076	0.99711 ± 0.00336	0.99858 ± 0.00057	0.97970 ± 0.01974	0.98088 ± 0.01065

improved and maintained high performance across all strategies. However, a closer look at the individual client-wise results in Table 12 reveals a subtle yet consistent trend: beyond 10 features, the F1-score begins to decline slightly for all strategies, including independent local training. This indicates a mild overfitting effect when the feature space grows too large.

Interestingly, despite this slight dip, the overall performance remains more stable compared to the earlier experiments performed with four clients. This suggests that the increased number of clients in the 6-client setting improved the model's ability to generalize and helped it withstand overfitting more effectively. The richer diversity of data across additional slices contributed to a more robust global model. While independent local training achieved similar performance with fewer features, FL began to outperform it with six or more features, consistent with the results observed in the Experiment 1.1

Taken together, the results across sections 5.1–5.3 support several generalizations of FLAccShield. The performance-weighted aggregation is validated under two structurally different learning paradigms, supervised classification and unsupervised anomaly detection, without task-specific hyperparameter tuning, supporting its applicability to FL settings beyond DDoS detection where clients produce heterogeneous local models. The stability advantage under non-IID conditions holds consistently across all clients and feature set sizes evaluated, and the 6-client experiment confirms the strategy scales without degradation when additional heterogeneous clients are introduced. However, all clients in the current experiments hold equal-sized datasets, the non-IID scenario is limited to attack-type partitioning, and evaluation is conducted on a single network traffic domain. Generalizations to unequal client dataset sizes, covariate shift in benign traffic, and other IDS datasets or FL application domains represents a natural direction for future investigation.

5.4 Ablation study

To isolate the contribution of each component of FLAccShield, we conduct an ablation study under the IID supervised setting at feature count 10, the configuration where baseline performance degradation is most visible in Experiment 1.1 (Table 9). We evaluate five configurations: (i) plain FedAvg, (ii) FedAvg augmented with performance weighting only (FedAvg+W), (iii) FedAvg augmented with the learning rate scheduler only (FedAvg+LRS), (iv) FedAvg augmented with two-level early stopping only (FedAvg+ES), and (v) the full FLAccShield combining all three components. Results are reported per client in Table 13.

5.4.1 Performance weighting

FedAvg+W achieves the largest gains over plain FedAvg on the underperforming clients C1 and C2, improving from 0.9642 to 0.9974 (+0.033) and from 0.9652 to 0.9979 (+0.033) respectively, while maintaining comparable performance on the already well-converged clients C3 and C4. This confirms that prioritizing low-accuracy clients during aggregation is the primary mechanism by which

Table 13 – Per-client F1 scores for ablation study (Feature count = 10, IID, supervised CNN)

Strategy	Feature count	Client 1	Client 2	Client 3	Client 4
FedAvg	10	0.964,2	0.965,2	0.999,3	0.999,3
FedAvg + W	10	0.997,4	0.997,9	0.999,2	0.999,2
FedAvg + LRS	10	0.994,4	0.994,0	0.999,2	0.999,2
FedAvg + ES	10	0.977,1	0.979,2	0.998,9	0.999,0
FLAccShield	10	0.978,4	0.981,8	0.999,0	0.999,0

FLAccShield corrects client-level performance imbalance, and establishes performance weighting as the dominant contributor among the three components.

5.4.2 Learning rate scheduler

FedAvg+LRS improves C1 and C2 to 0.9944 and 0.9940 respectively, a meaningful gain over plain FedAvg (+0.030 and +0.029) but below the improvement achieved by weighting alone. This confirms that the scheduler acts as a convergence stabilizer that reduces oscillation in later training stages, contributing positively but not as the primary driver of FLAccShield's performance gains.

5.4.3 Early stopping and full FLAccShield

FedAvg+ES and full FLAccShield produce lower C1 and C2 scores (0.9771/0.9784 and 0.9792/0.9818) than both FedAvg+W and FedAvg+LRS at feature count 10 under IID conditions. This is consistent with the design intent of early stopping: it is calibrated to prevent overfitting as feature count and data heterogeneity increase, and its regularisation effect introduces a conservative bias that marginally constrains peak performance when overfitting pressure is low. The benefit of early stopping is more pronounced in the non-IID setting (Section 5.1.2), where FLAccShield limits F1 degradation to ~7% as feature count grows from 4 to 10, compared to 10–22% for all baselines. Despite not matching the peak of FedAvg+W in isolation, full FLAccShield recovers the majority of FedAvg's degradation at feature count 10 (C1: 0.9642 → 0.9784, C2: 0.9652 → 0.9818) while simultaneously providing the convergence stability that weighting alone does not guarantee across all experimental conditions. The ablation results therefore support the design rationale of FLAccShield as a bundle: performance weighting drives accuracy recovery, while the convergence controls trade a small amount of peak IID performance for robustness under distribution shift and feature space growth.

6. CONCLUSION

This paper demonstrates how FL can be applied to detect DDoS attacks and anomalies in 5G network slices while ensuring strict privacy requirements. One of the key contributions of this paper is the proposal of a novel FL strategy, FLAccShield, which optimizes training time and helps maintain balanced performance across all clients. Evaluation under both IID and non-IID datasets, across a wide range of feature sets, has provided insights into how the model's performance varies under these conditions. For supervised training, we observed that the model maintained an F1 score of approximately 0.99 with only four features, while unsupervised training required 37 features to achieve an F1 score of 0.91. In the supervised setup, increasing the number of features beyond 4 led to overfitting, whereas, in unsupervised training, more features generally improved performance. This overfitting was more prominent in non-IID data than in IID data. These trends were consistent for the baseline strategies as well. In most of the conducted experiments, FLAccShield outperformed all other baseline strategies, and in some cases, it performed equally. Additionally, it consistently outperformed independent local training in most experiments, showing promising results for the future of FL in 5G security.

Despite the strong results, there are some limitations. Our non-IID dataset faced challenges due to the lack of subclass segregation for benign traffic. We also used HTTP to communicate between the client and server, which is insecure, making the model susceptible to potential interception or compromise of the training weights. Moreover, as there is only one server in the client-server architecture, it introduces the risk of a single point of failure. Additionally, as the FL server does not have control over local clients, any local data manipulation could harm the entire FL process. These are limitations of our work, but they open up avenues for future research.

In future work, we plan to acquire more realistic non-IID datasets and explore cryptographic techniques such as Secure Aggregation (SecAgg) to improve the security of weights in motion. Additionally, we will investigate Byzantine-resilient aggregation techniques and anomaly detection mechanisms to mitigate data poisoning issues.

We will also use HTTPS for communication between the client and server and explore peer-to-peer FL to avoid a single point of failure. Finally, to streamline the training process, we will look into integrating ML pipelines to automate and optimize the entire training process.

REFERENCES

[1] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. "5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15". In: *IEEE Access* 7 (2019), pp. 127639–127651. DOI: 10.1109/ACCESS.2019.2939938.

[2] Md Sajid Khan, Behnam Farzaneh, Nashid Shahriar, Niloy Saha, and Raouf Boutaba. "SliceSecure: Impact and Detection of DoS/DDoS Attacks on 5G Network Slices". In: *2022 IEEE Future Networks World Forum (FNWF)*. IEEE, 2022, pp. 639–642. DOI: 10.1109/FNWF55208.2022.00117.

[3] Mohamed Aly Bouke and Azizol Abdullah. "An empirical assessment of ML models for 5G network intrusion detection: A data leakage-free approach". In: *e-Prime - Advances in Electrical Engineering, Electronics and Energy* 8 (2024), p. 100590. ISSN: 2772-6711. DOI: https://doi.org/10.1016/j.prime.2024.100590. URL: https://www.sciencedirect.com/science/article/pii/S2772671124001700.

[4] Md Sajid Khan, Behnam Farzaneh, Nashid Shahriar, and Md Mahibul Hasan. *DoS/DDoS Attack Dataset of 5G Network Slicing*. 2023. DOI: 10.21227/32k1-dr12. URL: https://dx.doi.org/10.21227/32k1-dr12.

[5] Fahdah Alalyan, Mirna Awad, Wael Jaafar, and Rami Langar. "Secure Distributed Federated Learning for Cyberattacks Detection in B5G Open Radio Access Networks". In: *IEEE Open Journal of the Communications Society* (2024), pp. 1–1. DOI: 10.1109/OJCOMS.2024.3523468.

[6] Abdoul-Aziz Maiga, Edwin Ataro, and Stanley Githinji. "XGBoost and Deep Learning Based-Federated Learning for DDoS Attack Detection in 5G Core Network VNFs". In: *2024 6th International Conference on Computer Communication and the Internet (ICCCI)*. 2024, pp. 128–133. DOI: 10.1109/ICCCI62159.2024.10674312.

[7] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani. "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy". In: *2019 International Carnahan Conference on Security Technology (ICCST)*. 2019, pp. 1–8. DOI: 10.1109/CCST.2019.8888419.

[8] Chriki Zerrouk and Fatima Zohra. "Network Intrusion Detection System for Denial-of-Service Attack Detection in 5G". Master's Thesis. Barcelona, Spain: Facultat d'Informàtica de Barcelona (FIB), Universitat Politècnica de Catalunya (UPC) - BarcelonaTech, June 2024. URL: https://upcommons.upc.edu/bitstream/handle/2117/417190/188193.pdf.

[9] Rodrigo Moreira, Rodolfo S Villaça, Moisés RN Ribeiro, Joberto SB Martins, João Henrique Corrêa, Tereza C Carvalho, and Flávio de Oliveira Silva. "An intelligent native network slicing security architecture empowered by federated learning". In: *Future Generation Computer Systems* 163 (2025), p. 107537.

[10] Beibei Li, Yuhao Wu, Jiarui Song, Rongxing Lu, Tao Li, and Liang Zhao. "DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems". In: *IEEE Transactions on Industrial Informatics* 17.8 (2021), pp. 5615–5624. DOI: 10.1109/TII.2020.3023430.

[11] Md Sajid Khan. "Detection of DoS and DDoS Attacks on 5G Network Slices Using Deep Learning Approach". A Thesis Submitted to the Faculty of Graduate Studies and Research In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science, University of Regina. x, 85 p. Master's Thesis. Regina, Canada: University of Regina, 2024. URL: https://hdl.handle.net/10294/16534.

[12] Renato S. Silva, Carlos Colman Meixner, Rafael S. Guimarães, Thierno Diallo, Borja O. Garcia, Luís F. M. de Moraes, and Magnos Martinello. "REPEL: A Strategic Approach for Defending 5G Control Plane From DDoS Signalling Attacks". In: *IEEE Transactions on Network and Service Management* 18.3 (2021), pp. 3231–3243. DOI: 10.1109/TNSM.2020.3035342.

[13] Aditya Chilukuri, Shwetha Vittal, and A Antony Franklin. "SENTINEL: Self protecting 5G core control plane from DDoS attacks for high availability service". In: *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE. 2023, pp. 554–562.

[14] Ana Serrano Mamolar, Pablo Salvá-García, Enrique Chirivella-Perez, Zeeshan Pervez, Jose M. Alcaraz Calero, and Qi Wang. "Autonomic Protection of Multi-Tenant 5G Mobile Networks Against UDP Flooding DDoS Attacks". In: *Journal of Network and Computer Applications* 145 (2019), p. 102416. DOI: 10.1016/j.jnca.2019.102416.

[15] Huw Whitworth, Saba Al-Rubaye, Antonios Tsourdos, and Julia Jiggins. "5G Aviation Networks Using Novel AI Approach for DDoS Detection". In: *IEEE Access* 11 (2023), pp. 77518–77542. DOI: 10.1109/ACCESS.2023.3296311.

[16] Hubert Djuitcheu, Tirth Shah, Tammen Matthias, and Hans Dieter Schotten. "DDoS impact assessment on 5G system performance". In: (Sept. 2023). DOI: 10.36227/techrxiv.24072594.v1. URL: http://dx.doi.org/10.36227/techrxiv.24072594.v1.

[17] Sanfilippo, Salvatore. *hping3: A Network Scanning and Packet Crafting Tool*. http://www.hping.org/. Accessed: 2024-04-13. 2023.

[18] Oliveira, Jorge. *Low Orbit Ion Cannon (LOIC)*. https://github.com/NewEraCracker/LOIC. Accessed: 2024-04-13. 2023.

[19] Qinghang Gao, Hao Wang, Liyong Wan, Jianmao Xiao, and Long Wang. "G/M/1-Based DDoS Attack Mitigation in 5G Ultradense Cellular Networks". In: *Security and Communication Networks* 2022 (2022). Special Issue: Developments in Mobile Multimedia Technologies 2022, pp. 1–12. DOI: 10.1155/2022/4282859. URL: https://doi.org/10.1155/2022/4282859.

[20] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. "Federated learning for intrusion detection system: Concepts, challenges and future directions". In: *Computer Communications* 195 (2022), pp. 346–361.

[21] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. "A State-of-the-Art Survey on Solving Non-IID Data in Federated Learning". In: *Future Generation Computer Systems* 135 (2022), pp. 244–258. DOI: 10.1016/j.future.2022.05.003. URL: https://www.sciencedirect.com/science/article/pii/S0167739X22001686.

[22] Muhammad Babar, Basit Qureshi, and Anis Koubaa. "Review on Federated Learning for Digital Transformation in Healthcare Through Big Data Analytics". In: *Future Generation Computer Systems* 160 (2024), pp. 14–28. DOI: 10.1016/j.future.2024.05.046. URL: https://www.sciencedirect.com/science/article/pii/S0167739X24002784.

[23] Francesco Tusa and Stuart Clayman. "End-to-End Slices to Orchestrate Resources and Services in the Cloud-to-Edge Continuum". In: *Future Generation Computer Systems* 141 (2023), pp. 473–488. DOI: 10.1016/j.future.2022.11.026. URL: https://www.sciencedirect.com/science/article/pii/S0167739X22003971.

[24] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017, pp. 1273–1282.

[25] Zhiqiang Wei, Yulong Liu, Yifan Zhang, Xueqiang Chen, Yiqing Xu, and Haibo He. "An optimal federated learning-based intrusion detection for IoT networks". In: *Scientific Reports* 15.1 (2025), p. 10363. DOI: 10.1038/s41598-025-93501-8. URL: https://www.nature.com/articles/s41598-025-93501-8.

[26] Ansam Khraisat, Ammar Alazab, Sarabjot Singh, Tony Jan, and Alfredo Jr. Gomez. "Survey on Federated Learning for Intrusion Detection System: Concept, Architectures, Aggregation Strategies, Challenges, and Future Directions". In: *ACM Comput. Surv.* 57.1 (Oct. 2024). ISSN: 0360-0300. DOI: 10.1145/3687124. URL: https://doi.org/10.1145/3687124.

[27] William Marfo, Deepak K. Tosh, and Shirley V. Moore. "Network Anomaly Detection Using Federated Learning". In: *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*. 2022, pp. 484–489. DOI: 10.1109/MILCOM55135.2022.10017793.

[28] Nour Moustafa and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015, pp. 1–6. DOI: 10.1109/MilCIS.2015.7348942.

[29] Zeinab Zoghi and Gursel Serpen. "UNSW-NB15 Computer Security Dataset: Analysis Through Visualization". In: *arXiv preprint* (2021). arXiv: 2101.05067.

[30] Abdelwahab Boualouache and Thomas Engel. "Federated Learning-based Inter-slice Attack Detection for 5G-V2X Sliced Networks". In: *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. 2022, pp. 1–6. DOI: 10.1109/VTC2022-Fall57202.2022.10012736.

[31] Free5GC. *Free5GC: Open-Source 5G Core Network*. Accessed: 2025-03-29. 2025. URL: https://free5gc.org/.

[32] The TCPdump Group. *TCPdump: Command-Line Packet Analyzer*. Accessed: 2025-03-29. 2025. URL: https://www.tcpdump.org/.

[33] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In: *International Conference on Information Systems Security and Privacy*. 2018. URL: https://api.semanticscholar.org/CorpusID:4707749.

[34] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, and Mika Ylianttila. *5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network*. 2022. DOI: 10.21227/xtep-hv36. URL: https://dx.doi.org/10.21227/xtep-hv36.

[35] Behnam Farzaneh. "Deep Transfer Learning-Based DDoS Attack Detection in 5G and Beyond Networks". A Thesis Submitted to the Faculty of Graduate Studies and Research In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science, University of Regina. xvi, 74 p. Master's Thesis. Regina, Canada: University of Regina, 2024. URL: https://hdl.handle.net/10294/16533.

[36] CICFlowMeter Developers. *CICFlowMeter: An Intrusion Detection Flow Generator*. Accessed: February 11, 2025. 2023. URL: https://github.com/ahlashkari/CICFlowMeter.

[37] Gints Engelen, Vera Rimmer, and Wouter Joosen. "Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study". In: *2021 IEEE Security and Privacy Workshops (SPW)*. 2021, pp. 7–12. DOI: 10.1109/SPW53761.2021.00009.

[38] Wireshark Foundation. *Wireshark: The World's Most Popular Network Protocol Analyzer*. https://www.wireshark.org/. Accessed: 2024-04-13. 2023.

[39] Mahendra Prasad, Rahul Kumar Gupta, and Sachin Tripathi. "A Multi-level Correlation-Based Feature Selection for Intrusion Detection". In: *Arabian Journal for Science and Engineering* 47.8 (2022), pp. 10719–10729. ISSN: 2191-4281. DOI: 10.1007/s13369-022-06760-2. URL: https://doi.org/10.1007/s13369-022-06760-2.

[40] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. "Flower: A Friendly Federated Learning Research Framework". In: (2022). arXiv: 2007.14390 [cs.LG]. URL: https://arxiv.org/abs/2007.14390.

[41] Timothy O'Shea and Jakob Hoydis. "An Introduction to Deep Learning for the Physical Layer". In: *IEEE Transactions on Cognitive Communications and Networking* 3.4 (2017), pp. 563–575. DOI: 10.1109/TCCN.2017.2758370.

AUTHORS

Md Mahibul Hasan received the M.Sc. degree in Computer Science from the University of Regina, Canada, in 2025, under the supervision of Dr. Nashid Shahriar. Prior to his graduate studies, he worked as a wireless core engineer for five years. His research interests include privacy-preserving intrusion detection systems and federated learning for 5G network slices. His thesis proposed a federated learning-based collaborative intrusion detection framework for multi-slice 5G networks that preserves data privacy across participating nodes. He also investigated early-stage intrusion detection using only the first few packets of a network flow, work that was demonstrated at the University of Waterloo in collaboration with the Department of National Defence and presented at BSides Regina. He is a recipient of the 2024 Saskatchewan Innovation Award. He has also participated in ITU AI/ML in 5G Challenge competitions, where his team at the University of Regina placed second in detecting rare network intrusion types.

Aymen Ben Said received his MSc and PhD degrees in Computer Science from the University of Regina, Canada. His research interests include machine learning, federated learning, anomaly detection, artificial intelligence, combinatorial optimization, and scheduling, with applications in distributed learning and network security.

Karamveer Singh Sidhu is a MSc in Computer Science student at the University of Northern British Columbia, Canada. His work focuses on improving the security of Federated Learning, Deep Learning based Instrusion Detection Systems. He has previously published research into artificial intelligence applications in agriculture. He also works as a Software Engineer at Tokenbooks, where he develops full-stack systems for a Web3 finance platform.

Nashid Shahriar is an Associate Professor in the Department of Computer Science at the University of Regina (UofR), Canada. He received his PhD with the 2020 Alumni Gold Medal and the Mathematics Doctoral Prize from the School of Computer Science at the University of Waterloo. Dr. Shahriar's research has been recognized with multiple awards, including PST 2025 Best Paper Award, NOMS 2022 Best Student Paper Award, IM 2021 Best PhD Dissertation Award, CNSM 2019 Best Paper Award, NetSoft 2019 Best Student Paper Award, and CNSM 2017 Best Paper Award. He received 2025 Faculty of Science Award for Research Excellence at UofR and 2023 Young Professional Award from the IEEE Communications Society Technical Committee on Network Operation and Management. Dr. Shahriar is an Associate Editor of IEEE Transactions on Network and Service Management and has served on the TPC and OC of several international conferences. His research interests include network management, network security and reliability, and cybersecurity.

Sajal Saha (Senior Member, IEEE) is an Assistant Professor of Computer Science at the University of Northern British Columbia (UNBC), Canada, and Founder of the INFORM Lab. He received the Ph.D. degree in Computer Science from Western University, Canada, where he was nominated for the Governor General's Gold Medal, and the M.Sc. degree in Computer Science from Brock University, Canada, where he was recognized as a Distinguished Graduate. His research focuses on Internet traffic analysis, cyber-attack detection and mitigation, privacy-preserving and federated learning, and quantum-secure AI systems. Dr. Saha has authored numerous peer-reviewed publications, and his work has been supported through national and institutional research grants. He also serves as a reviewer and technical program committee member for international journals and conferences.