Federated Learning for 5G and Beyond, a Blessing and a Curse- An Experimental Study on Intrusion Detection Systems

Taki Eddine Toufik Djaidja[a,*], Bouziane Brik[a], Abdelwahab Boualouache[b], Sidi Mohammed Senouci[a] and Yacine Ghamri-Doudane[c]

[a]*DRIVE Laboratory EA1859, University of Burgundy, 49 Rue Mademoiselle Bourgeois, Nevers, 5800, France*
[b]*FSTM, University of Luxembourg, Luxembourg*
[c]*L3I Laboratory, Univ. La Rochelle, France*

ARTICLE INFO

Keywords:
Federated Learning
Deep Learning
IDS
NON-IID
5G and beyond

ABSTRACT

5G's service providers now leverage Deep Learning (DL) to automate their network slice management, provisioning, and security. To this end, each slice owner contributes data to feed a common dataset used to train centralized learning models. However, this method raises privacy considerations that prevent its usage. Therefore, Federated learning (FL), a collaborative approach that ensures data privacy, is being investigated while striving toward the same performance as centralized learning. As 5G and beyond services are so diverse, the local slice's data is not intended to reflect the entire data distribution. Thus, local data of slices are Non-Independently and non-Identically distributed (Non-IID), posing a challenge for FL-based models. In this paper, we investigate the use of FL to secure network slices and detect potential attacks. For that purpose, we first propose an architecture for deploying intrusion detection systems (IDSs) in 5G and beyond networks. Next, we thoroughly evaluate the latest state-of-art FL algorithms, including FedAvg, FedProx, FedPer, and SCAFFOLD, in the context of Independently and Identically Distributed (IID) and Non-IID data distributions. We compare these FL models to centralized and local DL models. We find that SCAFFOLD outperforms all the other FL algorithms and ensures a stable learning loss convergence, a promising finding that strengthens the case for leveraging FL in IDS development. Nevertheless, none of the FL models could achieve the centralized model's performance in Non-IID scenarios.

1. Introduction

5G and beyond networks' ambition is to build an intelligent connected environment that spans communications on the ground, in the air, and the space (Chowdhury, Shahjalal, Ahmed and Jang, 2020). Indeed, 5G networks pave the way for a wide range of applications, including vehicular networks, unmanned aerial vehicles, smart homes, real-time streaming, and virtual and augmented reality (Dogra, Jha and Jain, 2021). 5G's adoption of the network slicing paradigm allows the creation of logical, self-contained networks known as slices. Each business client (service provider) will have its slice(s) with independent control and management, facilitating the on-demand creation of network resources. In other terms, the conventional, one-size-fits-all components of cellular networks are being replaced by softwarized and virtualized components that may accommodate diverse network requirements, depending on the delivered service (Zhang, 2019). The security of the network and its end-users must be considered for any new communication technology; fortunately, both industry and academia are designing countermeasures and safeguards to address these concerns. Network protocols, firewalls, and IDSs are examples of these technical solutions (Nencioni, Garroppo and Olimid, 2021). These safeguards complement each other; for example, IDSs supplement firewalls by capturing intrusions targeting network protocols and applications.

Nowadays, most of the research community's efforts are directed toward creating IDSs that focus on protection against any potential intrusion. The consideration of IDS began to acquire popularity in the early 2000's, and regained interest in 2015 with the advent of Machine Learning (ML) and, in particular, DL (Lavaur, Pahl, Busnel and Autrel, 2022).
In this new wave of IDSs, patterns of network intrusions are learned from available intrusion datasets using DL techniques, and once the DL models are deployed, they are used to analyze and identify intrusions. Current DL-based IDSs rely on network traffic traces, which are aggregated to network flows, including meta-data header information and statistical data of packets in the flows (Hofstede, Čeleda, Trammell, Drago, Sadre, Sperotto and Pras, 2014). The flows are then labeled as benign or malicious, as well as the type of malicious activity. For that purpose, DL models collect data (labeled network flows) from all involved parts and network slices in the context of 5G and beyond networks, constructing a global dataset including all the flows that are used in the training phase to provide a global model; this method is known as centralized training.

Industries and service providers in 5G and beyond networks must manage their networks and design security

*Corresponding author
taki-eddine.djaidja@u-bourgogne.fr (T.E.T. Djaidja); bouziane.brik@u-bourgogne.fr (B. Brik); abdelwahab.boualouache@uni.lu (A. Boualouache); sidi-mohammed.senouci@u-bourgogne.fr (S.M. Senouci); yacine.ghamri@univ-lr.fr (Y. Ghamri-Doudane)
ORCID(s): 0000-0003-4131-5343 (T.E.T. Djaidja); 0000-0002-3267-5702 (B. Brik); 0000-0001-6237-6597 (A. Boualouache); 0000-0001-8525-9596 (S.M. Senouci); 0000-0002-7986-2476 (Y. Ghamri-Doudane)

mechanisms for their network slices. In practice, however, the centralized approach has proven inconceivable because the network flows dataset contains sensitive information about companies and slice owners. This raises concerns about business privacy, labeled datasets can disclose that an attack has occurred within the network slice, and companies are reluctant to share this information. Therefore, sharing the data to build a global dataset is resisted by individual slice owners. Even so, network slices must collaborate to deploy effective IDSs; unfortunately, DL models trained independently at the slice level using local slice data may miss relevant samples of malicious and benign traffic, resulting in a decrease in the effectiveness of the learning model.

To deal with this challenge, the community has been considering distributed techniques to perform training without requiring data sharing across slices. The advent of FL (McMahan, Moore, Ramage, Hampson and Arcas, 2016), a framework that allows decentralized training, has been a catalyst for collaborative IDS research (Lavaur et al., 2022). FL consists of performing local training for different clients, with the clients communicating just their local model parameters to a coordinator. This approach ensures privacy since local data is not shared. The coordinator performs an aggregation procedure, then this aggregated model is returned to clients; the training process is repeated until convergence. In (McMahan et al., 2016), the federated average (FedAvg) aggregation function was introduced; it simply averages the clients' model weights in the aggregation phase to obtain the global model at each training round. However, FL faces many challenges that impede its use in the industry, including concerns about statistical heterogeneity (non iid-ness), communication efficiency, and security and privacy (Mothukuri, Parizi, Pouriyeh, Huang, Dehghantanha and Srivastava, 2021). The authors in (Li, Sahu, Talwalkar and Smith, 2020; Rahman, Ahmed, Akhter, Hasan, Amin, Aziz, Islam, Mukta and Islam, 2021; Wahab, Mourad, Otrok and Taleb, 2021; Ma, Zhu, Lin, Chen and Qin, 2022) reviewed these challenges and possible solutions found in the literature.

To recap, the main objective of FL is to deliver a DL model with similar performances to the centralized one. Statistical heterogeneity concern compromises this objective, and the FedAvg aggregation function has shown its limitation in this realistic scenario, suffering from a weight divergence from an optimal model that leads to unstable convergence (Karimireddy, Kale, Mohri, Reddi, Stich and Suresh, 2019). In light of these challenges, new aggregation methods have been introduced, including Federated with the Proximal term (FedProx) (Li, Sahu, Zaheer, Sanjabi, Talwalkar and Smith, 2018) and SCAFFOLD (Karimireddy et al., 2019) as two novel aggregation techniques that solve the high weight divergence caused by Non-IID data by using controlling mechanisms in client updates. The Federated Personalizing (FedPer) (Arivazhagan, Aggarwal, Singh and Choudhary, 2019) aggregation approach tackles the issue by personalizing DL models via transfer learning.

To contribute to the development of FL-based IDS capable of addressing the aforementioned issue, this study investigates FL and the aggregation methods by which it can be utilized to enable collaborative DL-based IDS in a 5G and beyond slicing environment. This paper aims to delve into state-of-the-art FL algorithms, namely FedProx, FedPer, and SCAFFOLD, designed to address the challenge of weight divergence in heterogeneous networks. It focuses on two primary settings: one where data is uniformly distributed across network slices (IID), and another, more realistic scenario where data distribution is non-uniform (Non-IID) among slices. In this context, the paper explores situations where malicious traffic may manifest in one slice while being absent in another. The paper also conducts a comparative analysis of the performance of these models in comparison to locally trained models and centralized global models, a comparison that has been overlooked in existing research works. Furthermore, within this paper, we present a framework for the training and deployment of FL-based IDSs within the context of 5G and beyond networks. The experimental study reveals that FL can be viewed as a "curse" because the exchanging models involve communication overhead but do not improve detection accuracy and performance compared to locally trained models. Fortunately, state-of-art approaches could overcome convergence problems in Non-IID data, a "blessing" for future FL-based models to enable effective distributed learning.

The rest of the paper is organized as follows. Section 2 explains the FL aggregation algorithms while it also covers the deployment of intrusion detection systems in 5G and beyond networks; Section 3 overviews the related works regarding the use of FL in securing 5G networks. Section 4 outlines our methodology, details the used dataset and its partitioning for the IID/Non-IID scenarios, and specifies the DL models, including centralized, FL, and local models. The experimental evaluations of these models are presented in section 5. Finally, Section 6 discusses the work and brings the study to a close.

2. Background

This section provides general background on FL and its various aggregation algorithms. Furthermore, we describe IDSs.

2.1. Federated Learning

FL is a distributed DL approach to build a single global model from data stored on multiple entities; these entities are referred to as clients, which represent network slices in our context. The model training is conducted at the client level, specifically within each 5G network slice; only model weights are exchanged across clients. The learning of FL was formulated in (McMahan et al., 2016) as a minimization function:

$$\min_{w} f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \ , \ F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w) \quad (1)$$

Where K is the number of clients, $f_i(w) = l(x_i, y_i; w)$ represents the loss function of input i where the output is predicted by the model having $W = w$, F_k is the objective function for client k, P_k is the client's k local dataset with $n_k = |P_k|$ and $n = \sum n_k$.

FL was initially designed to reduce the communication overhead (McMahan et al., 2016). This communication efficiency is achieved by sharing model weights rather than client-generated large-sized raw data. Additionally, FL can guarantee more data privacy since the client's local data will always reside at the client level. However, one of the primary objectives of FL is to show classification/prediction results comparable to models trained in a centralized fashion. This can be challenging because a single set of client data is not supposed to reflect the data distribution in the domain.

Starting from an initial DL model, the process begins by training separate models locally for a certain number of iterations E, each client using its data. These respective models are then communicated to a central node, and aggregated using an aggregation function to obtain a global model. This process is called a round. At the end of each round, the clients receive the aggregated model, and the process is repeated for a certain number of rounds R. The majority of FL models rely on client-server architecture, in which the server (or coordinator) is in charge of initializing the model and performing aggregation at each round. Other FL topologies, such as peer-to-peer architecture, are also possible.
Algorithms 1, 2 summarize the FL training procedure for both the server-side (coordinator-) and client-sides, respectively. Regarding the client procedure (Algorithm 2), η represents the learning rate, $\nabla l(b; w)$ is the gradient of the loss function ($f_i(w)$).

In the next sub-sections, we will explore four aggregation algorithms, namely FedAvg (McMahan et al., 2016), FedProx (Li et al., 2018), FedPer (Arivazhagan et al., 2019), and SCAFFOLD (Karimireddy et al., 2019).

Algorithm 1 FL Server (Coordinator)

```
Input: clients // list of $k$ client
Input: $r$
Output: $w_r$
1: $w_0 \leftarrow$ init()
2: for $t = 1$ to $r$ do
3:     $w_t^k \leftarrow$ clients.local_train($t$,$w_{t-1}$) // Executing Algo. 2
       for all $k$ clients
4:     $w_t \leftarrow$ aggregate($w_t^k$)
5: end for
6: return $w_r$
```

Algorithm 2 FL Client Local Train

```
Input: $t$,$w_{t-1}$,$P_k$,$E$,$batch\_size$
Output: $w_t^k$
1: $\beta \leftarrow$ getBatches($P_k$,$batch\_size$)
2: $w_t^k \leftarrow w_{t-1}$
3: for $e = 1$ to $E$ do
4:     for $b$ in $\beta$ do
5:         $w_t^k \leftarrow w_t^k + \eta \nabla l(b; w_t^k)$
6:     end for
7: end for
8: return $w_t^k$
```

2.1.1. *FedAvg*

Federated average is the first averaging algorithm proposed in (McMahan et al., 2016). The approach consists of simply averaging the weights of the different models communicated by the clients. As shown in (Equation 2), the global model w_t at round t is calculated by averaging $n_k w_t^k$, this line will replace the line 4 in algorithm 1.

$$w_t \leftarrow \frac{1}{n} \sum_{k=1}^{K} n_k w_t^k \quad (2)$$

2.1.2. *FedProx*

FedProx (Li et al., 2018) was suggested to deal with heterogeneous networks; seeking to encounter FedAvg's limitations in Non-IID situations. FedProx changes the objective function on the client side, by adding a proximal term (μ) to regulate the direction of weights at the client level, and prevent the client's model from deviating from the global model communicated, by the coordinator at the beginning of the round. The line 5 in algorithm 2 will be:

$$w_t^k \leftarrow w_t^k + \eta(\nabla l(w_t^k, b) + \frac{\mu}{2} ||w_t^k - w_{t-1}||^2) \quad (3)$$

Where w_{t-1} is the global aggregated model of the previous round ($t-1$), μ is the proximal term and $||w_t^k - w_{t-1}||^2$ is an L2 norm which measures the distance between the model during the local training and the previously communicated global model.

2.1.3. *SCAFFOLD*

SCAFFOLD (Karimireddy et al., 2019) stands for the stochastic controlled averaging algorithm. Its main aim is to reduce the client variance induced by heterogeneity in client updates. In this approach, both the clients and server (coordinator) procedures are updated and are shown in algorithms 3 and 4. The server's model and all the clients' models are equipped with control variates, c and c_k, respectively. The difference (c_k-c) estimates the client (k)'s weight drift and is used to control the client update (line 6 in algorithm 4).

2.1.4. *FedPer*

FedPer (Arivazhagan et al., 2019) is a transfer learning-inspired approach. It involves modifying the training process of FL by integrating transfer learning technique. This

Algorithm 3 FL SCAFFOLD Server

```
Input: clients,$r$
Output: $w_r$
1: $w_0 \leftarrow$ init()
2: $c_0 \leftarrow 0$
3: for $t = 1$ to $r$ do
4:     $(\Delta w_t^k, \Delta c_t^k) \leftarrow$ clients.local_train($t$,$w_{t-1}$,$c_{t-1}$)
5:     $(\Delta w_t, \Delta c_t) \leftarrow \frac{1}{K} \sum_{k=1}^{K} (\Delta w_t^k, \Delta c_t^k)$
6:     $w_t \leftarrow w_{t-1} + \gamma \Delta w_t$
7:     $c_t \leftarrow c_{t-1} + \Delta c_t$
8: end for
9: return $w_r$
```

Algorithm 4 FL SCAFFOLD Client Local Train

```
Input: $t$,$w_{t-1}$,$c_{t-1}$,$P_k$,$E$,$batch\_size$
Output: $\Delta w_t^k$,$\Delta c_t^k$
1: if $(t = 1)$ then $c_0^k \leftarrow 0$ // Init
2: $\beta \leftarrow$ getBatches($P_k$,$batch\_size$)
3: $w_t^k \leftarrow w_{t-1}$
4: for $e = 1$ to $E$ do
5:     for $b$ in $\beta$ do
6:         $w_t^k \leftarrow w_t^k + \eta(\nabla l(b; w_t^k) - c_{t-1}^k + c_{t-1})$
7:     end for
8: end for
9: $c_t^k \leftarrow \nabla l(w_{t-1}, P_k)$
10: $(\Delta w_t^k, \Delta c_t^k) \leftarrow (w_t^k - w_{t-1}, c_t^k - c_{t-1}^k)$
11: return $\Delta w_t^k$,$\Delta c_t^k$
```

approach is compatible with various FL aggregation algorithms, including FedAvg and FedProx. In federated personalizing, the model is divided into the following: base and personalized layers. Only the base layers are communicated and aggregated in the server, while the personalized layers are trained locally; the architecture is shown in Fig. 1. Consequently, the FedPer aggregation updates the algorithm executed at the client level; at the first round, the clients initialize their local weights, and during the training, the used model is the concatenation of base and personalized layers. The algorithm 5 showcases the procedure.

2.2. Intrusion Detection Systems

Intrusion detection is the process of searching for indicators of attacks. Current research on intrusion detection is supported by the use of statistics, data mining, ML and DL algorithms, which can be used in supervised, semi-supervised, or unsupervised ways. These models are commonly referred to as data-driven models, and network flows are commonly used to train them.

A network flow is a grouping of packets identified by the tuple <source, source port, destination, destination port,protocol> (Hofstede et al., 2014). A flow is formed when the first packet arrives and is terminated by receiving a termination flag or exceeding a timeout. Secondly, network flows are enriched with statistical information such as flow time, the number of packets, packet length, and flag information. Finally, in an IDS that uses supervised learning, flows are labeled. The label can be normal or malicious, as well as the type of malicious activity in this case.

The following section will review recent research efforts that have employed ML/DL techniques for intrusion detection, with a particular focus on FL-based IDS approaches.

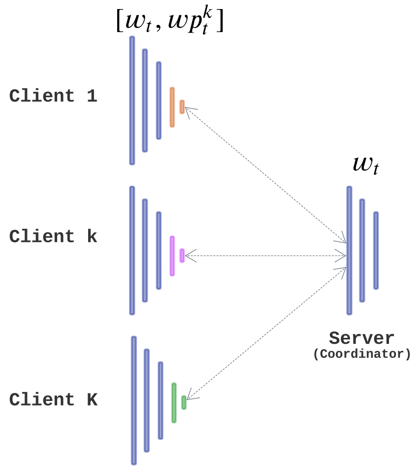

Figure 1: FedPer aggregation algorithm: layers in blue are the base layers w_t, and the other colors represent the personalized layers wp_t^k. The server knows only base layers.

Algorithm 5 FL FedPer Client Local Train

```
Input: $t$,$w_{t-1}$,$P_k$,$E$,$batch\_size$
Output: $w_t^k$
1: if $(t = 1)$ then
2:     $wp_0^k \leftarrow init()$
3: end if
4: $\beta \leftarrow$ getBatches($P_k$,$batch\_size$)
5: $temp_t^k \leftarrow (w_{t-1}, wp_{t-1}^k)$
6: for $e = 1$ to $E$ do
7:     for $b$ in $\beta$ do
8:         $temp_t^k \leftarrow temp_t^k + \eta \nabla l(b; temp_t^k)$
9:     end for
10: end for
11: $(w_t^k, wp_t^k) \leftarrow temp_t^k$
12: return $w_t^k$
```

3. Related Works

The appearance of FL in 2018 stimulated research efforts in distributed IDS (Lavaur et al., 2022). This section provides a summary of some recent works closely related to this topic.

In (Mothukuri, Khare, Parizi, Pouriyeh, Dehghantanha and Srivastava, 2022), the authors have proposed a Recurrent Neural Network (RNN) trained in a federated fashion to detect anomalies in Internet of Things (IoT) networks using the ModBus network dataset (Rysavy and Matousek, 2021). Seven different models were trained while varying the window size of their time-series data; then, these models were combined using a decision tree vote-based scheme to classify the traffic with high confidence.

Attack detection in wireless edge-enabled networks using FL while minimizing communication costs and ensuring high detection performance was discussed in (Chen, Lv, Liu, Fang, Chen and Pan, 2020). The authors leveraged attention mechanisms applied to RNN-based DL models in their FL training process. The use of attention is meant to increase the weight of important devices (clients) in the training phase.

In (Zhang, He, Ma, Gao, Ma and Avestimehr, 2021), the authors proposed an FL-based framework for intrusion detection in the IoT context. They used a semi-supervised scheme, where auto-encoder-based models are trained using the FedAvg algorithm across different IoT devices. The authors presented a method for calculating a global reconstruction error threshold for the traffic classification task. The authors used the NB-IoT dataset (Meidan, Bohadana, Mathov, Mirsky, Shabtai, Breitenbacher and Elovici, 2018) for the experiments, where data is collected from 9 IoT devices attacked with mainly the same attacks. The FL approach outperformed the local approach while having nearly the same performance as the centralized one, thanks to the introduced global reconstruction error threshold approach.

The authors in (Fan, Li, Zhan, Cui and Zhang, 2020) expressed the need of personalizing distributed DL-based model, in the context of 5G IoT, due to IoT devices' heterogeneity. The authors presented a similar approach to FedPer, by leveraging personalized layers, that are learned via transfer learning. At each training round, each client performs a train with a publicly shared dataset and communicates the model weights to the server to perform the aggregation (FedAvg). Then, the server returns the aggregated model to the clients, and each client performs a training epoch with its local data, on its personalized layers. The process is repeated for all training rounds. Furthermore, by utilizing a public dataset, that is shared by all clients, the approach is somehow inspired by knowledge distillation in FL. Four separate clients are simulating four different networks for the experimental phase; the authors used both IoTDataset (Kang, Ahn, Lee, Yoo, Park and Kim, 2019) and NSLKDD (Tavallaee, Bagheri, Lu and Ghorbani, 2009) datasets for the clients' data and CICIDS (Sharafaldin, Habibi Lashkari and Ghorbani, 2018) dataset for the public dataset. The authors revealed the good results obtained by their approach; nevertheless, the generalization aspect of each separate model is missing, as well as a comprehensive comparison with models trained on local data only.

The authors in (Rahman, Tout, Talhi and Mourad, 2020) investigated the implementation of DL-based IDS models, whether centralized, local, or based on FL. They evaluated the NSLKDD dataset in IID and Non-IID partitioning settings. The authors reported that FedAvg obtained comparable accuracy to the centralized approach while outperforming the on-device models. Besides, under Non-IID settings, FedAvg was marginally higher than on-device and significantly exceeded by the centralized approach.

These studies have proposed FL-based approaches for intrusion detection in cellular networks, demonstrating the superior performance of FL models over centralized and on-device models (Mothukuri et al., 2022; Chen et al., 2020). Conversely, in our study, the centralized model is assumed to be the optimal model with the best performances, and we are attempting to get similar results using FL. In addition to the investigation (Rahman et al., 2020), our work includes a pathological Non-IID scenario as well as multi-class DL model classifiers. Furthermore, we investigate various FL aggregation algorithms that have been proposed to address the limitations of FedAvg in heterogeneous networks.

4. Methodology

In this section, we detail our methodology to study the FL-enabled IDS performance involving the use of DL, FL aggregation algorithms, and the partitioning of IID and Non-IID datasets. Before we proceed, we first introduce the sliced 5G architecture as well as the system model we considered in this study.

4.1. Intrusions and DL based Intrusion Detection in 5G and beyond networks

In our given scenario, a 5G network slice provides a service, such as video streaming or Internet of Things (IoT) application. This service is deployed on the Application Server (AS), a network function belonging to the network slice. End users of the slice access and utilize this service by establishing communication with the AS. However, a potential security threat arises when an internal node, assumed to be an attacker, attempts to compromise the AS by sending malicious network traffic.

In the context of DL-based IDS in 5G and beyond networks, the Network Data Analytics Function (NWDAF) -which is a network function that supports the deployment of DL models pipeline in the 5G network(Yuan, Gehrmann, Sternby and Barriga, 2022)- gathers network packets from the User Plan Function (UPF) network function that serves as a gateway in the 5G context, and, as previously mentioned (sub-section 2.2), transforms these network packets into network flows. The flow is then analyzed using the DL model, and the classification result is communicated to the Security NF, in order to devise the appropriate policy, for example, in the event of an attack.

For supervised DL training of IDS, it is essential for the involved parts to reach a consensus on certain hyperparameters. These hyperparameters include the choice of the DL architecture, class labels, and information regarding normalization. Subsequently slice's security engineers gather elaborated flows from the NWDAF; the training is then performed after data labeling. The DL model training can be performed locally or collaboratively with other slices, which include centralized and federated methods. Finally, the DL model is deployed at the NWDAF level, and the SecurityVNF will request the NWDAF to classify future network traffic. Fig. 2 depicts and illustrates the whole process.

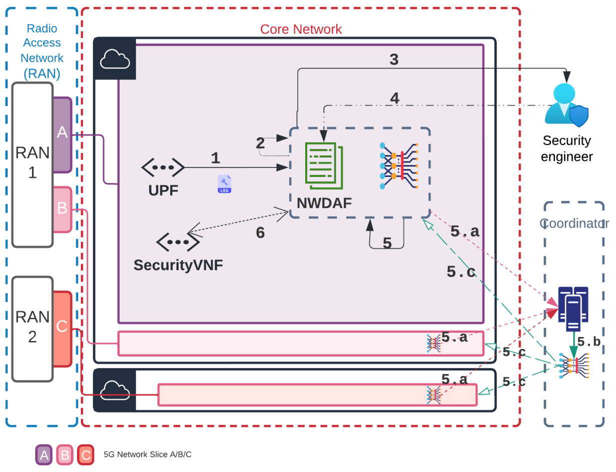

Figure 2: DL based IDS in 5G and Beyond : 1. NWDAF collects traffic from UPF, 2. NWDAF generates network flows, 3. Security engineer gathers the unlabeled flows, 4. Security engineer labels the flows and sends them back to NWDAF, 5. Training phase: (*i*) Local training is done at the NWDAF level, (*ii*) Centralized training: labeled flows are transmitted to the coordinator (5.a), which performs training (5.b) and returns the final model (5.c), (*iii*) FL: for each round, model weights are transmitted to the coordinator (5.a), which performs aggregation (5.b) and returns the aggregated model (5.c), 6. SecurityVNF requests the NWDAF when new traffic arrives

We consider a set of K slices (or clients following the FL process), each slice c_k has a labeled flow P_k; where $P_k = Tr_k \cup Te_k$, Tr_k and Te_k refer to train and test set for client k, respectively. We also define $M_{\text{centralized}}$ as the centralized model, which has $Tr_{\text{centralized}}$ and $Te_{\text{centralized}}$ as training and test sets; where $Tr_{\text{centralized}} = Tr_1 \cup Tr_2 \cup \cdots \cup Tr_K$ and $Te_{\text{centralized}} = Te_1 \cup Te_2 \cup \cdots \cup Te_K$. For FL approaches, slices will act as clients, and we build three FL global models using three different aggregation algorithms (M_{fedavg}, M_{fedprox} and M_{scaffold}), and K personalized model M^k_{fedper}. Besides, M^k_{local} reflects the local models trained on Tr_k and Te_k. In order to evaluate the different DL models M_x, we also consider a validation dataset. We evaluate the performance of the models (M_x) with this validation set, assumed as new traffic to evaluate future use.

4.2. Dataset Description and Pre-processing

In our study, we used NSLKDD dataset (Tavallaee et al., 2009), since it was widely used by the research community, and is considered as a reference dataset for flow-based intrusion detection. While the dataset itself was not originally generated within a 5G context, it is noteworthy that the simulated communication between a user and a server in the dataset can be assumed to occur within a 5G infrastructure, and the data is captured at the level of the UPF.

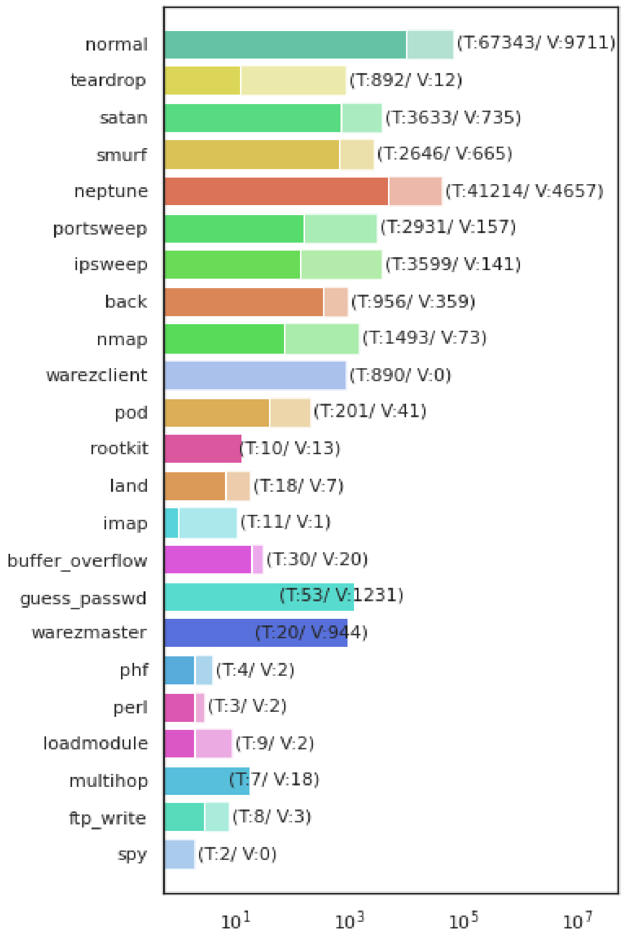

Figure 3: NSLKDD label distribution T: Train / V : Validation datasets

The primary data files of NSLKDD are "*KDDTrain+*" and "*KDDTest+*". The former will be used during the training phase this include train and test data for the different models, while the latter will be kept for the validation phase. The purpose of utilizing the validation dataset is to evaluate the performance of the various models using previously unseen data. The results related to this validation data are depicted in Figs. 8 and 12.

The training dataset has 148,517 records with 41 features and covers 23 different labels. The dataset is mostly made up of normal and Neptune attack traffic, the latter is a flooding-based denial-of-service (DoS) attack.

Fig. 3 displays the label distribution of NSLKDD dataset. Features' nature is either Boolean, real, or categorical. In regard to data pre-processing, we used min-max normalization to convert real values to a range from 0 to 1. In our contribution, we assumed that the involved network slices communicate their minimum local values to enable global min-max normalization.

The dataset has three categorical features: protocol type, service, and flag. The categorical features are encoded between 0 and 1 using the following encoding process, e.g.

the protocol type has three values: 'tcp', 'udp', and 'icmp' and the corresponding value for each category equals to its order count divided by the number of categories (3), given that the order of counting starts from zero. Hence, the result is $\{0, \frac{1}{3}, \frac{2}{3}\}$. Flow labels are indeed encoded using one-hot encoding; at this stage, we can expect the DL model to have 41 input and 23 output sizes.

The "*KDDTest+*", utilized as the validation dataset in our research, captured new types of attacks. However, our study is constrained to the labels already present in the training set, and these new attack labels are consequently excluded from the validation set. While new attacks are advantageous for binary classification (normal and abnormal), enabling researchers to assess the efficiency of their IDS models by training them to recognize normal behavior and label any new attack patterns as abnormal, our study employs a multi-class supervised learning approach to identify the specific type of attack. For novel attack labels, however, we cannot make predictions for these since we lack that knowledge during the training phase.

The training dataset is partitioned into K subsets for the generating slices' (clients') data P_k. As explained in the previous sub-section, each P_k is further divided into training and testing sets for the training phase of the DL models. Meanwhile, the validation dataset V is reserved exclusively for the final validation of the DL models. Partitioning scenarios are discussed in the next subsection.

4.3. Partitioning Scenarios

In our study, we consider eight slices ($K = 8$), where "*KDDTrain+*" dataset is partitioned according to two different scenarios. We chose this number of clients ($K = 8$) due to practical reasons. The NSLKDD dataset, although valuable, isn't very large. Including more clients could have made the dataset less representative. Thus, we decided that having eight clients would provide a manageable dataset size, allowing us to explore diverse scenarios effectively.
The labels distribution for both scenarios are shown in Figs. 4 and 5.

4.3.1. IID Data-based scenario

In this scenario (Fig. 4), the "*KDDTrain+*" data is randomly partitioned across the clients using a uniform distribution, assuming that each slice has known more or less the same nature of traffic, hence the slices undergo the same attacks and with the same frequency.

4.3.2. Non-IID Data-based scenario

This pathological Non-IID scenario (Fig. 5), is more realistic, where the assumption that slices undergo the same type of attacks is not valid. So, in this scenario, we assume that normal traffic is more or less collected with the same distribution across slices. However, for attacks, we assume that the traces of some attacks are only available in one slice and not in others.

Table 1
Neural Network (M_x Architecture)

N	Layers	Size	
0	Linear	41	**input layer**
1	Linear ReLu BatchNorm	128 128	
2	Linear ReLu BatchNorm	256 256	
3	Linear ReLu BatchNorm	128 128	
3	Linear ReLu BatchNorm	128 128	
4	Linear ReLu BatchNorm	64 64	
5(*)	Linear ReLu BatchNorm	32 32	
6(*)	Linear ReLu DropOut	23	**output layer**

4.4. Neural Network Architecture and Training

Table 1 summarizes the neural network architecture, which is used to build all the models M_x described previously.

We opted for a Multi-Layer Perceptron (MLP) architecture, which is appropriate for our specific context, where we treat each network flow as a single data point. We believe that in other contexts where the data structure is different (such as time series data), other DL architectures like RNNs and Transformers may prove to be more suitable. Our DL model has six feed-forward layers, each followed by a ReLu activation function and a batch normalization layer, while the output layer is succeeded by a dropout layer. In the FedPer approach (M^k_{fedper}), the layers 5(*) and 6(*) are used as personalized layers as detailed in sub-section 2.1.4.

The majority of the rows in the training sets (Tr_k and $Tr_{\text{centralized}}$) are normal and neptune, as seen in Figs. 4 and 5. To handle this unbalanced data situation, we used weighted CrossEntropy Loss for the DL training function. We used Stochastic Gradient Descent (SGD) as the learning optimizer, setting the learning rate (η) at $3e^{-4}$. Concerning the remaining training-related hyper-parameters, we set the batch size to 64, the training rounds (R) to 200, the number of local epochs learning (E) to 1, the proximal term for FedProx (μ) to 0.3 and the step size for SCAFFOLD (γ) to 1. These hyper-parameters were selected following a series of tests using a generate-and-test methodology for hyper-parameter tuning.

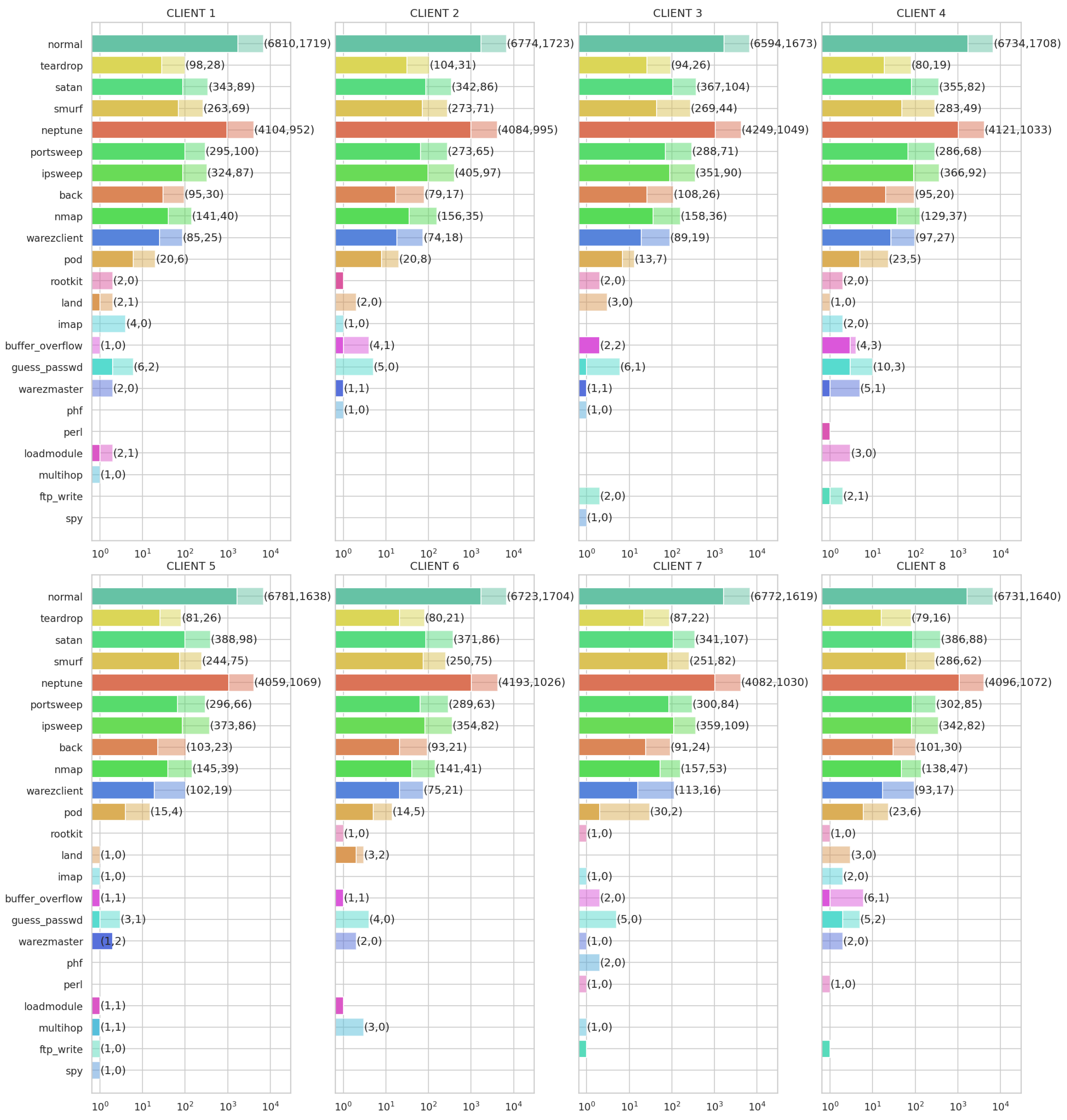

Figure 4: IID Scenario Label distribution over K clients, (Tr_k, Te_k)

5. Experimental Study and Results

This section presents the experimental study and evaluates the performances of different DL models M_X on top of both IID and Non-IID scenarios (sub-sections 5.1 and 5.2 respectively) and different aggregation algorithms. We conducted the model training and experiments using the PyTorch[1] framework on a system with the configuration: Intel Core i7-10700, 32GB RAM, Nvidia RTX 3070.

5.1. IID-based Scenario

Fig. 6 depicts the training classification loss, the weighted cross entropy as mentioned previously, during each training round/epoch of the different DL models (M_X) for IID scenario, with the lines and dashed lines in the graph

[1]pytorch.org/

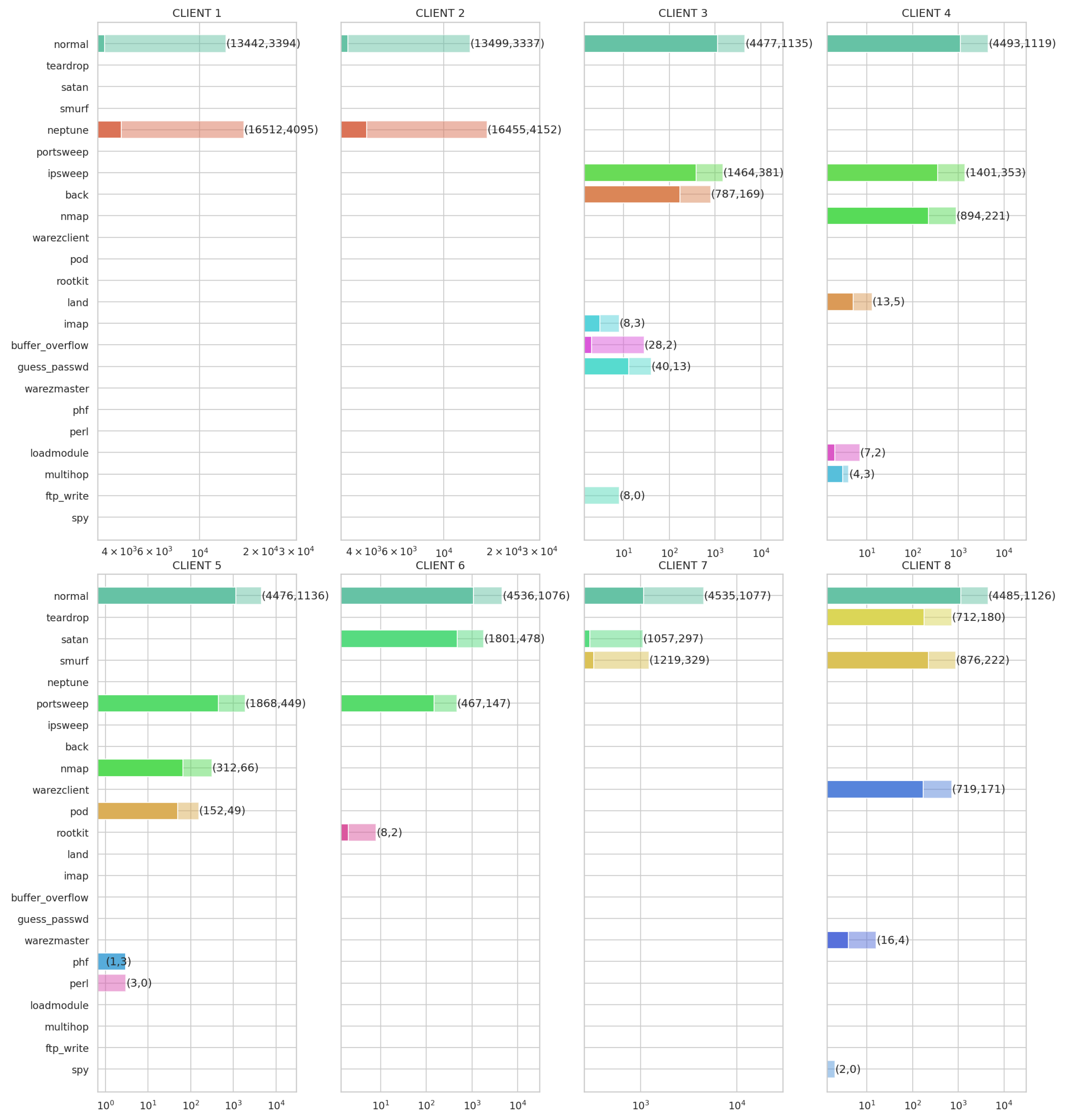

Figure 5: Non-IID Scenario Label distribution over K clients, (Tr_k,Te_k)

representing the average loss on the training (Tr_k) and test (Te_k) sets at each round/epoch, respectively. Furthermore, the colored contour shows the confidence interval of the train loss average. It is calculated as the average plus-minus the standard deviation of the loss in (Tr_k); this confidence interval presents insight about the loss trend across all clients. The four FL models in this scenario converge at the same pace, which is promising. Another interesting aspect is that the confidence interval is so tight, that the loss standard variation is $\approx$ 0. This implies that the loss value is approximately the same for all clients at each round, simply due to the similar distribution in this training scenario. We also remark that both centralized and local models converge at around the 25th epoch. But, they then started to over-fit the training data set, due to over-training of the model, which led to over-fitting of the common labels (normal and Neptune). Another reason is the

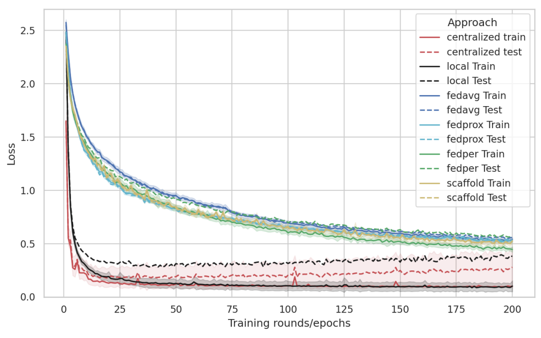

Figure 6: IID LOSS.

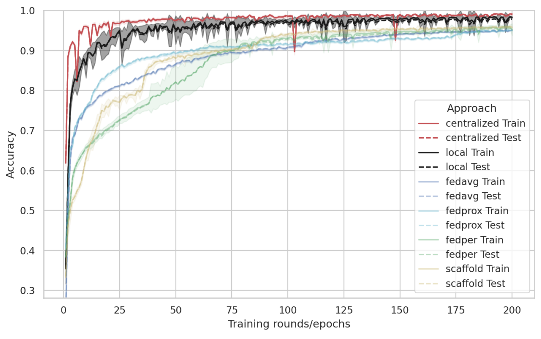

Figure 7: IID Accuracy.

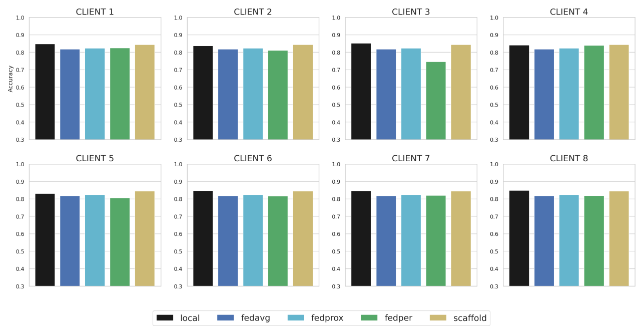

Figure 8: IID Accuracy on validation dataset.

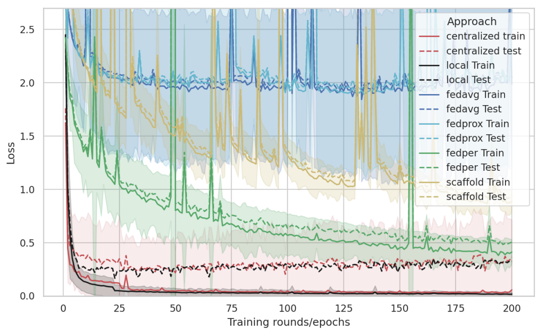

Figure 9: Non-IID LOSS

weighted loss function, which penalizes the error on the non-dominant labels more. Furthermore, the centralized model outperforms the other models, despite the fact that they all have the same distribution. This is because the centralized model has a complete view of all the distinct traces of the different labels.

Fig. 7 displays the M_X's accuracy during the training process. We clearly observe that both the centralized and local models provide the best results. The previously mentioned minor over-fitting between train and test does not appear, because the accuracy calculation is not weighted, and non-dominant labels have the least impact on this metric. In addition, most of the federated methods progress toward the same acceptable accuracy of 95 ($\approx$ 95%); FedPer and SCAFFOLD converge the slowest, however after a few rounds (25), they start to follow the converging tendency like other FL approaches.

The models' efficiency is evaluated by showing how they perform on a validation dataset. Accuracy in validation data is shown in Fig. 8. The accuracy values of the global models ($M_{\text{centralized}}$, M_{fedavg}, M_{fedprox} and M_{scaffold}) are identical throughout all clients, while for FedPer and local models (M^k_{fedper}, M^k_{local}) they are proper to each client k. We can see that all models achieved the same performances ($\approx$ 80%). The federated techniques were able to deliver similar results as the centralized approach, which is the purpose of FL. However, the same outcomes were also obtained by local training, putting into question the necessity of collaboration in this scenario.

5.2. Non-IID-based Scenario

Similarly to the previous scenario, we elaborate on the same evaluation/analysis for the Non-IID scenario. The (M_X)s' train/test loss is presented in Fig. 9. The average loss (represented in the Fig. by the lines and dashed lines) for FedAvg and FedProx stays unchanged at ($\approx$ 2) and does not decrease. Secondly, after 100 rounds, these two approaches' average standard deviation values are ($\approx$ 0.8), indicating a large confidence interval. This signifies that the classification loss is not uniform across clients, with certain clients achieving good error scores, while others do not. Meanwhile, SCAFFOLD shows lower train/test loss rates ($\approx$ 1) compared to FedProx and FedAvg. Additionally, it has the narrowest confidence interval, with an average standard deviation of ($\approx$ 0.17), indicating that M_{scaffold} achieves almost the same error rate for all clients.

Fig. 10 displays the training loss over the K clients for FedProx and SCAFFOLD and clearly illustrates these findings. For SCAFFOLD, we can see that the loss is descending equitably (to $\approx$ 1) for all clients, but FedProx is converging just for clients 1 and 2. However, the loss sticks and does

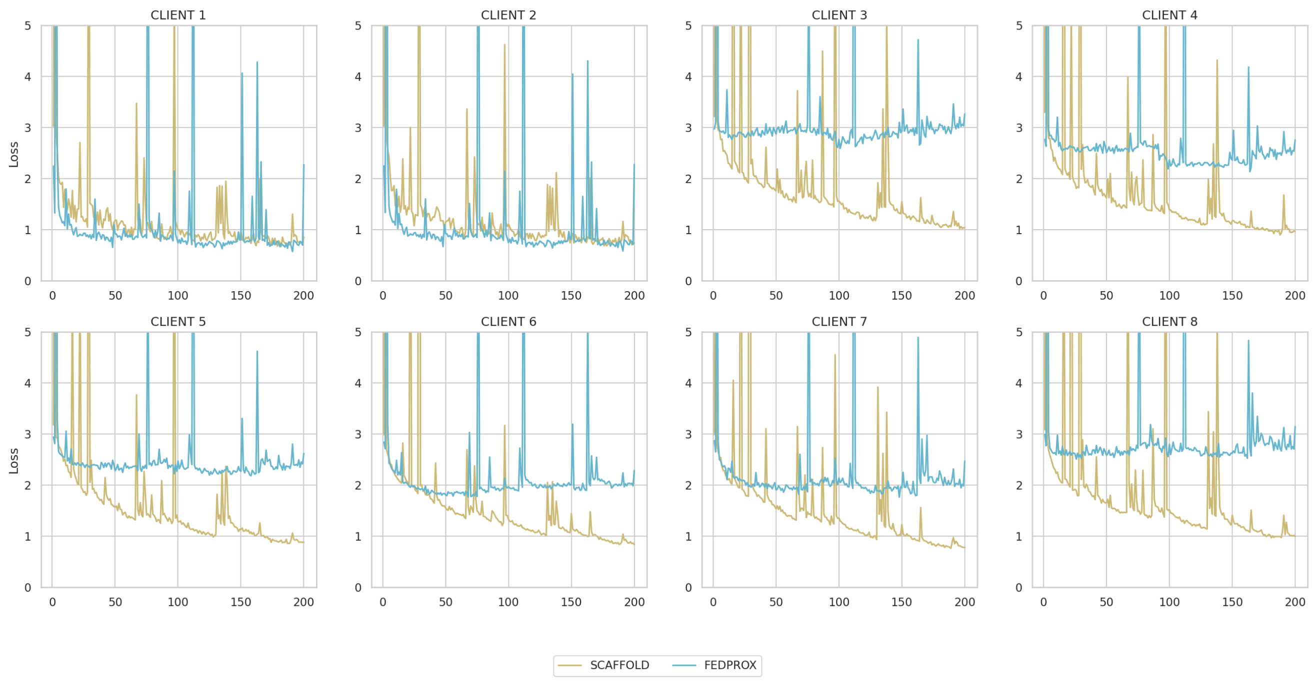

Figure 10: SCAFFOLD Vs FedProx loss

not decrease for other clients, indicating poor learning. Both clients 1 and 2 contain normal and Neptune labels; FedProx is forced to converge to a model that fits these two labels while diverging from the optimal model that is expected to learn all labels.

FedPer has the lowest loss when compared to other FL approaches. Each personalized model achieves the minimum loss on its train/test data, with an average loss of less than 0.5. These values were not previously obtained with other FL models. However, before the validation process, it is too early to make any conclusions. Local models also provide good results, with a narrower confidence interval than the centralized approach. This is because each client has a limited number of labels, which makes it easier for M^k_{local} to learn labels of Tr_k/Te_k. For example, client 1 must learn just normal and Neptune. Also, for the same reasons indicated in the IID experiments (see subsection 5.1), the centralized model produces the best performance.

The accuracy graph (Fig. 11) aligns with the loss graph. The accuracy of centralized and local models is close to 100%. Moreover, an interesting observation is that FedPer is showing comparable performances. FedAvg and FEDProx have a large confidence interval and an accuracy of approximately (80%); while SCAFFOLD reaches (90%) accuracy with a small confidence interval, it outperforms both FedAvg and FedProx.

In the next paragraphs, we present and discuss the validation phase. We first investigate personalized models (FedPer and local models) and as shown in Fig. 12, M^k_{fedper} and M^k_{local} models achieve the same accuracy rates over all clients K. $M^{1,2}_{\text{fedper}}$ and $M^{1,2}_{\text{local}}$ models learn well the Neptune attack since it is frequent in $Tr_{1,2}$, but also in the validation dataset. This explains why $M^{1,2}_{\text{fedper}}$ and $M^{1,2}_{\text{local}}$ have high accuracy rates on the validation data ($\approx 70\%$). Nonetheless, we notice that employing FedPer does not lead in any knowledge sharing between the different M^k_{fedper}, as seen by poor accuracy rates for clients $c_3..c_8$. We find that the personalized layers (wp^k) trained via transfer learning force the M^k_{fedper} model to generate a model similar to the local one (M^k_{local}). This finding also explains why FedPer performs better across the different Tr_k/Te_k; each personalized model M^k_{fedper} has lost its generalization ability and is over-fitting its locally known labels.

Table 2 contains the validation results for the remaining global models, including the true positives (TP), true positives rate (TPR), positive predictive value (PPV) aka precision, and F1 score metrics for each label in the validation dataset; please refer to (Naser and Alavi, 2021) for further details on these metrics. The three FL models perform well in learning the normal and Neptune labels, but only FedAvg and SCAFFOLD can do so for the Nmap label (label 3 in Table 2). SCAFFOLD outperforms FedAvg and FedProx for labels 4 to 7. For example, SCAFFOLD can efficiently learn the Pod attack, which is not frequent in the training dataset. However, labels 8-13 are unfortunately not learned by the FL models; while the centralized model identifies those labels adequately, the FL models ignore them. This demonstrates the current limitations of FL approaches to deal with pathological Non-IID contexts.

Meanwhile, the centralized model fails to detect labels 14-20, which may be due to various reasons, including the

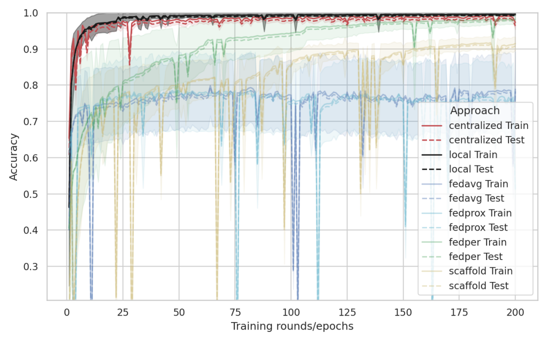

Figure 11: Non-IID Accuracy

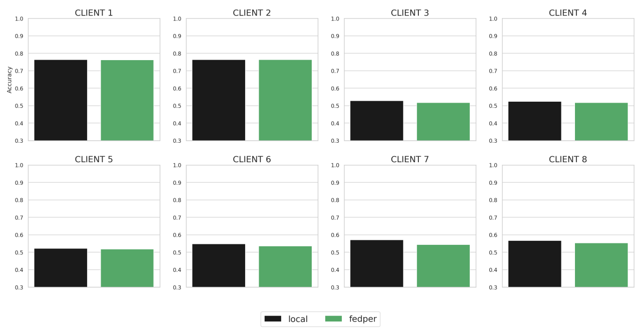

Figure 12: Local and FedPer accuracy on validation dataset in Non-IID scenario.

pre-processing and encoding processes, the quantity/quality of training data, the DL architecture, etc. However, this study aims to compare the effectiveness of FL models with the centralized approach. The centralized approach is assumed to be optimal with satisfactory results (labels 1-13). In the previous paragraphs, we showed that SCAFFOLD outperformed the other FL-based aggregation approaches and maybe a solution for a stable convergence to an optimal model (which could be similar to the centralized one). Even so, further enhancement of the control mechanisms is required to overcome SCAFFOLD's limitations (e.g. labels 8-13).

6. Discussion and Conclusion

FL is a key enabler in 5G and future networks, intending to allow cognitive intelligence across various slices while maintaining the privacy of each slice's data. FedAvg was the first proposed method that showed limitations when dealing with Non-IID data. As a result, several techniques were designed, notably FedPer, FedProx, and SCAFFOLD.

We showed that for IID scenario, FL algorithms performed effectively. However, similar results were also obtained when each slice trains a model with only its local data. In reality, uniform sampling gives the same label distribution throughout slices, resulting in equivalent local and federated training models. The efficacy of FL, consequently, cannot be measured in this scenario.

In the second scenario, we considered a more realistic scenario in which the label distribution is not identical across FL clients (Non-IID scenario). We revealed that FedAvg converged for a limited number of clients' data while diverging in others, asserting that the model was forced to learn only a specific set of labels while ignoring the others. This results in poor loss decreasing during training. FedProx surpassed FedAvg; however, FedProx, like FedAvg, has the same previous issue even though it has a controlling mechanism to deal with it. SCAFFOLD was the only global model that provided stable convergence across the FL clients; SCAFFOLD outperformed other FL approaches.

We also showed that FedPer training via transfer learning tends to replicate local models and over-fit the local data. This approach is not yet mature since it loses generalization potential. Despite this, we can envision FedPer being a solution in cases where the same sample has different client labels. This requires further investigation to locate these reflective labels and perform transfer learning to encounter this situation.

While classifying new attack traffic, which was not previously known (referred to as 'unknown unknowns'), remains a challenge for deep learning-based models when trained in a centralized manner, FL currently faces difficulties in sharing knowledge to classify the 'known unknowns.' It struggles to deliver a global model capable of providing similar capabilities to centralized models due to statistical heterogeneity.

We believe that the current limitations related to statistical heterogeneity of FL in enabling cognitive knowledge in realistic scenarios extend beyond intrusion detection applications. They may also manifest in other areas relying on FL in 5G and beyond networks, including edge and fog computing.

These challenges pose potential obstacles to the realization of self-X networks, where future networks must autonomously plan, configure, manage, optimize, and self-heal. Therefore, to effectively address the issue of statistical heterogeneity in FL, it is imperative to allocate additional research efforts toward enhancing FL aggregation methodologies. One promising avenue involves exploring control systems similar to SCAFFOLD. Investigating control mechanisms for FL and developing sophisticated control strategies that adapt to the dynamic nature of FL systems can help maintain model stability and prevent issues related to model weight divergence. Another promising approach is to delve into metalearning within the context of FL. Metalearning equips FL models with the ability to quickly adapt to new environments, making them more versatile and efficient. This adaptability is particularly valuable in 5G and beyond networks characterized by heterogeneous conditions.

In this paper, we suggest an architecture for IDSs in the context of multi-slice 5G and beyond networks, discuss

Table 2
Detection metrics for FedAVG,FedProx,SCAFFOLD and Centralised models

	label	N	FAVG				FPROX				SCA				CEN			
			TP	TPR	PPV	F1	TP	TPR	PPV	F1	TP	TPR	PPV	F1	TP	TPR	PPV	F1
1	normal	9711	9645	0.99	0.72	0.83	9642	0.99	0.71	0.83	9623	0.99	0.72	0.84	9199	0.95	0.82	0.88
2	neptune	4657	1457	0.31	1.00	0.48	1507	0.32	0.97	0.49	1426	0.31	1.00	0.47	4567	0.98	1.00	0.99
3	nmap	73	71	0.97	0.66	0.79	0	0.00	/	/	67	0.92	0.28	0.43	73	1.00	0.55	0.71
4	satan	735	397	0.54	0.68	0.60	372	0.51	0.85	0.63	490	0.67	0.82	0.74	457	0.62	0.74	0.68
5	prtswep	157	90	0.57	0.03	0.06	99	0.63	0.26	0.37	119	0.76	0.61	0.67	144	0.92	0.41	0.56
6	pod	41	0	0.00	/	/	0	0.00	/	/	25	0.61	0.60	0.60	35	0.85	0.73	0.79
7	land	7	3	0.43	0.33	0.38	1	0.14	0.08	0.10	7	1.00	0.00	0.01	5	0.71	1.00	0.83
8	smurf	665	0	0.00	0.00	/	0	0.00	/	/	8	0.01	0.53	0.02	656	0.99	0.97	0.98
9	back	359	2	0.01	0.67	0.01	0	0.00	0.00	/	0	0.00	/	/	324	0.90	0.58	0.71
10	ipswep	141	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/	138	0.98	0.93	0.96
11	tear	12	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/	11	0.92	0.21	0.34
12	phf	2	0	0.00	/	/	0	0.00	0.00	/	0	0.00	/	/	1	0.50	0.14	0.22
13	perl	2	0	0.00	0.00	/	0	0.00	/	/	0	0.00	/	/	1	0.50	0.09	0.15
14	rtkit	13	0	0.00	/	/	0	0.00	/	/	0	0.00	/	/	0	0.00	0.00	/
15	imap	1	0	0.00	/	/	0	0.00	0.00	/	0	0.00	/	/	0	0.00	0.00	/
16	buffer	20	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/
17	guess	1231	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/
18	mlthop	18	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/	1	0.06	0.06	0.06
19	ldmdle	2	0	0.00	/	/	0	0.00	0.00	/	0	0.00	0.00	/	0	0.00	0.00	/
20	ftp	3	0	0.00	/	/	0	0.00	/	/	0	0.00	0.00	/	0	0.00	0.00	/

and experiment with various state-of-the-art FL aggregation approaches to intrusion detection, and finally, emphasize the limitations of FL as it stands at present.

Acknowledgments

This work was supported by the 5G-INSIGHT bilateral project (ID: 14891397) / (ANR-20-CE25-0015-16), funded by the Luxembourg National Research Fund (FNR), and by the French National Research Agency (ANR).

CRediT authorship contribution statement

Taki Eddine Toufik Djaidja: Conceptualization of this study, Methodology, Software, Validation, Investigation, Writing - Original Draft. **Bouziane Brik:** Supevision, Study and methodology Validation, writing and proofreading. **Abdelwahab Boualouache:** Supevision, Study and methodology Validation, writing and proofreading. **Sidi Mohammed Senouci:** Supevision, Study and methodology Validation, writing and proofreading. **Yacine Ghamri-Doudane:** Supevision, Study and methodology Validation.

References

Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S., 2019. Federated learning with personalization layers. URL: https://arxiv.org/abs/1912.00818, doi:10.48550/ARXIV.1912.00818.

Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K., Pan, W., 2020. Intrusion detection for wireless edge networks based on federated learning. IEEE Access 8, 217463–217472. doi:10.1109/ACCESS.2020.3041793.

Chowdhury, M.Z., Shahjalal, M., Ahmed, S., Jang, Y.M., 2020. 6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions. IEEE Open Journal of the Communications Society 1, 957–975. doi:10.1109/OJCOMS.2020.3010270.

Dogra, A., Jha, R.K., Jain, S., 2021. A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies. IEEE Access 9, 67512–67547. doi:10.1109/ACCESS.2020.3031234.

Fan, Y., Li, Y., Zhan, M., Cui, H., Zhang, Y., 2020. Iotdefender: A federated transfer learning intrusion detection framework for 5g iot, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), pp. 88–95. doi:10.1109/BigDataSE50710.2020.00020.

Hofstede, R., Čeleda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A., 2014. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. IEEE Communications Surveys & Tutorials 16, 2037–2064. doi:10.1109/COMST.2014.2321898.

Kang, H., Ahn, D.H., Lee, G.M., Yoo, J.D., Park, K.H., Kim, H.K., 2019. Iot network intrusion dataset. URL: https://dx.doi.org/10.21227/q70p-q449, doi:10.21227/q70p-q449.

Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T., 2019. Scaffold: Stochastic controlled averaging for federated learning. URL: https://arxiv.org/abs/1910.06378, doi:10.48550/ARXIV.1910.06378.

Lavaur, L., Pahl, M.O., Busnel, Y., Autrel, F., 2022. The evolution of federated learning-based intrusion detection and mitigation: a survey. IEEE Transactions on Network and Service Management , 1–1doi:10.1109/TNSM.2022.3177512.

Li, T., Sahu, A.K., Talwalkar, A., Smith, V., 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine 37, 50–60. doi:10.1109/MSP.2020.2975749.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2018. Federated optimization in heterogeneous networks. URL: https://arxiv.org/abs/1812.06127, doi:10.48550/ARXIV.1812.06127.

Ma, X., Zhu, J., Lin, Z., Chen, S., Qin, Y., 2022. A state-of-the-art survey on solving non-iid data in federated learning. Future Generation Computer Systems 135, 244–258. URL: https://www.sciencedirect.com/science/article/pii/S0167739X22001686, doi:https://doi.org/10.1016/j.future.2022.05.003.

McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y., 2016. Communication-efficient learning of deep networks from decentralized data URL: https://arxiv.org/abs/1602.05629, doi:10.48550/ARXIV.1602.05629.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y., 2018. N-baiot: Network-based detection of iot botnet attacks using deep autoencoders.

Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehghantanha, A., Srivastava, G., 2022. Federated-learning-based anomaly detection for iot security attacks. IEEE Internet of Things Journal 9, 2545–2554. doi:10.1109/JIOT.2021.3077803.

Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G., 2021. A survey on security and privacy of federated learning. Future Generation Computer Systems 115, 619–640. URL: https://www.sciencedirect.com/science/article/pii/S0167739X20329848, doi:https://doi.org/10.1016/j.future.2020.10.007.

Naser, M.Z., Alavi, A.H., 2021. Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. Architecture, Structures and Construction URL: https://doi.org/10.1007%2Fs44150-021-00015-8, doi:10.1007/s44150-021-00015-8.

Nencioni, G., Garroppo, R.G., Olimid, R.F., 2021. 5g multi-access edge computing: Security, dependability, and performance. URL: https://arxiv.org/abs/2107.13374, doi:10.48550/ARXIV.2107.13374.

Rahman, K.M.J., Ahmed, F., Akhter, N., Hasan, M., Amin, R., Aziz, K.E., Islam, A.K.M.M., Mukta, M.S.H., Islam, A.K.M.N., 2021. Challenges, applications and design aspects of federated learning: A survey. IEEE Access 9, 124682–124700. doi:10.1109/ACCESS.2021.3111118.

Rahman, S.A., Tout, H., Talhi, C., Mourad, A., 2020. Internet of things intrusion detection: Centralized, on-device, or federated learning? IEEE Network 34, 310–317. doi:10.1109/MNET.011.2000286.

Rysavy, O., Matousek, P., 2021. Modbus dataset for ics anomaly detection. URL: https://dx.doi.org/10.21227/e1bc-3w91, doi:10.21227/e1bc-3w91.

Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization, pp. 108–116. doi:10.5220/0006639801080116.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. doi:10.1109/CISDA.2009.5356528.

Wahab, O.A., Mourad, A., Otrok, H., Taleb, T., 2021. Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Communications Surveys & Tutorials 23, 1342–1397. doi:10.1109/COMST.2021.3058573.

Yuan, Y., Gehrmann, C., Sternby, J., Barriga, L., 2022. Insight of anomaly detection with nwdaf in 5g, in: 2022 International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1–6. doi:10.1109/CITS55221.2022.9832914.

Zhang, S., 2019. An overview of network slicing for 5g. IEEE Wireless Communications 26, 111–117. doi:10.1109/MWC.2019.1800234.

Zhang, T., He, C., Ma, T., Gao, L., Ma, M., Avestimehr, S., 2021. Federated learning for internet of things, in: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, New York, NY, USA. p. 413–419. URL: https://doi.org/10.1145/3485730.3493444, doi:10.1145/3485730.3493444.

Taki Eddine Toufik DJAIDJA is a PhD student in the DRIVE Laboratory at the University of Burgundy since 2020. He received the BSc. Degree in Information Systems and Software Engineering and the MSc. Degree in Intelligent Computing Systems from the University of Science and Technology Houari Boumediene (USTHB) - Algeria, in 2015 and 2020, respectively. His main research is focused on the application of ML/DL techniques to future network automation. His research interests also include 5G and beyond, network slicing, network security, edge computing, cloud computing, and CDN.

BOUZIANE BRIK (IEEE Senior Member) received the Engineering degree (First Class) in computer science and the Magister degree from the University of Laghouat, Algeria, in 2010 and 2013, respectively, and the Ph.D. degree from the University of Laghouat, France, and the University of La Rochelle, France, in 2017. He is currently working as an Associate Professor with Burgundy (Bourgogne) University and DRIVE Laboratory. Before joining Burgundy University, he was a Postdoctoral Fellow with the University of Troyes, CESI School, and Eurecom School. His research interests also include the Internet of Things (IoT), the IoT in industrial systems, smart grid, and vehicular networks.

Dr. **Abdelwahab Boualouache** is a research associate at the Faculty of Science, Technology, and Medicine (FSTM), University of Luxembourg. His current research covers security and privacy in mobile and wireless networks, mainly focusing on topics related to 5G and beyond 5G cellular networks and connected and automated vehicles.

Sidi-Mohammed Senouci received his Ph.D. degree in computer science from the University of Paris XI and his HDR from the National Polytechnic Institute of Toulouse, France. From 2004 to 2010, he was a researcher with France Telecom R&D (Orange Labs), Lannion, France. Since 2010, he has been a professor at the Institut supérieur de l'automobile et des transports, Nevers, France, where he directs the DRIVE Laboratory. He holds seven international patents and has published his work in major conference proceedings and re - nowned journals.

Yacine Ghamri-Doudane received an engineering degree in computer science from the National Institute of Computer Science (INI), Algiers, Algeria, in 1998, an M.S. degree in signal, image and speech processing from the National Institute of Applied Sciences (INSA), Lyon, France, and a Ph.D. degree in computer networks from the Pierre & Marie Curie University, Paris 6, France, in 1999 and 2003, respectively. He is currently a full professor at the University of La Rochelle and director of its Laboratory of Informatics, Image and Interaction. His current research interests lie in the area of wireless networking and mobile computing, with a current emphasis on topics related to IoT, wireless sensor networks, and vehicular networks.