StatAvg: Mitigating Data Heterogeneity in Federated Learning for Intrusion Detection Systems

Pavlos S. Bouzinis[†], Panagiotis Radoglou-Grammatikis[‡§], Ioannis Makris[†], Thomas Lagkas[¶], Vasileios Argyriou[‖], Georgios Th. Papadopoulos[**], Panagiotis Sarigiannidis[‡], and George K. Karagiannidis[††]

***Abstract*—Federated learning (FL) enables devices to collaboratively build a shared machine learning (ML) or deep learning (DL) model without exposing raw data. Its privacy-preserving nature has made it popular for intrusion detection systems (IDS) in the field of cybersecurity. However, data heterogeneity across participants poses challenges for FL-based IDS. This paper proposes statistical averaging (`StatAvg`) method to alleviate non-independently and identically (non-iid) distributed features across local clients' data in FL. In particular, `StatAvg` allows the FL clients to share their individual local data statistics with the server. These statistics include the mean and variance of each client's feature vector. The server then aggregates this information to produce global statistics, which are shared with the clients and used for universal data normalization, i.e., common scaling of the input features by all clients. It is worth mentioning that `StatAvg` can seamlessly integrate with any FL aggregation strategy, as it occurs before the actual FL training process. The proposed method is evaluated against well-known baseline approaches that rely on batch and layer normalization, such as `FedBN`, and address the non-iid features issue in FL. Experiments were conducted using the TON-IoT and CIC-IoT-2023 datasets, which are relevant to the design of host and network IDS, respectively. The experimental results demonstrate the efficiency of `StatAvg` in mitigating non-iid feature distributions across the FL clients compared to the baseline methods, offering a gain in IDS accuracy ranging from 4% to 17%.**

***Index Terms*—cybersecurity, intrusion detection systems, federated learning, data heterogeneity, statistical averaging**

I. Introduction

In the dynamic era of smart technologies, including the Internet of Things (IoT) [1], artificial intelligence (AI) [2] and future wireless networks [3], [4], the attack surface increases significantly. In particular, from single-step attacks, the attackers now have the ability to design and execute multi-step attack scenarios, targeting multiple systems and domains in a coordinated and synchronized manner. According to the MITRE ATT&CK framework, notable examples of attack campaigns include (a) C0034 - the 2022 Ukraine Electric Power Attack, and (b) C0022 - Operation Dream Job. These incidents highlight the increasingly complex and evolving nature of cyber threats, which continue to pose significant risks to critical infrastructure and organizational security. As cyberattack strategies evolve and grow more complex, it is evident that traditional methods are no longer sufficient to safeguard critical infrastructure and organizational security. Hence, there is now a strong demand for reliable, real-time intrusion detection systems (IDS). In a cybersecurity landscape where threats can quickly morph and adapt, efficient IDS are not just important, but essential.

Traditionally, IDS rely on signature-based methods, where predefined attack rules or patterns, referred to as signatures, are identified and compared with the monitoring data, thus alerting a potential threat if a match is found. For instance, `Snort` and `Suricata` are popular IDS in this category. On the other hand, in recent years, both machine learning (ML) and deep learning (DL) models have already demonstrated significant promise as a means to detect cyberattacks [5]. However, it is worth mentioning that these models need the presence of appropriate security datasets that are often not publicly available, especially for critical domains [5]. In addition, appropriate adjustments are required to re-train and integrate these models. Finally, conventional ML/DL methods are conducted in a centralized fashion, where a central entity collects all the necessary data from endpoints to construct training datasets and afterwards generates the ML/DL models. Although this approach successfully enables the detection of intrusions, it raises privacy concerns since endpoints' private data are shared with third parties.

To alleviate privacy issues and mitigate communication overhead, federated learning (FL) has been proposed as an inherently privacy-preserving decentralized learning solution [6], [7]. According to the FL principles, the participating clients are building an ML/DL model collaboratively with the aid of a central entity (e.g., a central server). The salient

[*] *This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070450 (`AI4CYBER`). Disclaimer: Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.*

[†] P. S. Bouzinis and I. Makris are with MetaMind Innovation P.C., Kila Kozani, 50100, Kozani, Greece - `E-Mail: pbouzinis@metamind.gr; makris@metamind.gr`

[‡] P. Radoglou-Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, 50100, Kozani, Greece - `E-Mail: pradoglou@uowm.gr; psarigiannidis@uowm.gr`

[§] P. Radoglou-Grammatikis is also with K3Y Ltd, William Gladstone 31, 1000, Sofia, Bulgaria - `E-Mail: pradoglou@k3y.bg`

[¶] T. Lagkas is with the Department of Computer Science, Democritus University of Thrace, Kavala Campus, 65404, Kavala, Greece - `E-Mail: tlagkas@cs.duth.gr`

[‖] V. Argyriou is with the Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, London, UK - `E-Mail: vasileios.argyriou@kingston.ac.uk`

[**] G. Th. Papadopoulos is with the Department of Informatics and Telematics, Harokopio University of Athens, Omirou 9, Tavros, GR17778, Athens, Greece - `E-Mail: g.th.papadopoulos@hua.gr`

[††] G. K. Karagiannidis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece - `E-Mails: geokarag@auth.gr`

feature of FL is that clients transmit locally trained models to the server rather than raw data. Afterwards, the server aggregates the received parameters, updates the global model, and subsequently broadcasts it to the clients. Consequently, the server has no access to clients' raw datasets. However, despite the benefits of FL, a notable challenge in the design of an FL-based IDS is the existence of non-independently and identically distributed (iid) data among clients, commonly referred to as data heterogeneity. In particular, if the data is not representative across the clients, the global model may become biased, thus working efficiently on some cases but inaccurately on others. Also, the presence of non-iid data can affect the federated training procedure in terms of delaying or hindering convergence.

A. Related Work

The related work is organized into two subsections. The first focuses on the design of general FL-based IDS, while the second examines studies that address non-IID data challenges in FL-based IDS. The latter will primarily be the driving factor for the motivation behind our proposed approach.

1) FL-based IDS: Several works investigate the role and impact of FL in cybersecurity and, more precisely, in the scope of intrusion detection. Some survey papers in this field are listed in [8]–[12]. In [8], the authors present a comprehensive survey regarding the impact of FL within the scope of intrusion detection, highlighting challenges and future directions. In [9], a detailed comparison regarding centralized, distributed and FL-driven intrusion detection mechanisms for IoT environments was provided. Similarly, the authors in [10] discuss advances of FL within cybersecurity applications in IoT ecosystems. [11], provide a comprehensive study regarding FL-driven intrusion detection, game theory, social psychology and explainable AI. Finally, in [12], the authors focus their attention on security and privacy issues regarding FL applications. Next, we further discuss recent works that deliver FL-driven IDS.

In [13], the authors introduce `DeepFed`, an FL-driven IDS for cyber-physical systems. In this method, a trust authority is introduced, whose role is to produce the encryption keys for the proposed Pailier public-key cryptosystem utilized for the communication between the server and the industrial clients. For the detection process, the authors leverage a combined convolutional neural network (CNN) - gated recurrent unit (GRU), while special attention is paid to the proposed Pailier-based secure communication protocol for the communication between the server and the clients. Finally, three evaluation metrics are considered, namely accuracy, precision, recall and F1-score, demonstrating the detection efficiency of `DeepFed`.

In [14], authors describe `MV-FLID`, a multi-view FL-based IDS which focuses on the detection of attacks against message queuing telemetry transport (MQTT) communications within IoT environments. In particular, MV-FLID adopts a multi-view approach, combining bi-directional flow features, undirectional flow features and packet features. An FL model is generated for each of the previous viewpoints. Regarding the feature selection process, the authors leverage the grey wolf optimizer introduced in [15]. Next, the federated training procedure follows. Finally, an ensemble-based technique is used to combine the outcomes of the FL models in order to provide a unified prediction outcome. Traditional performance evaluation metrics are considered to demonstrate the overall detection effectiveness of `MV-FLID`.

In [16], a semisupervised FL scheme for intrusion detection within IoT environments was introduced. The proposed scheme relies on CNN models, while four phases are followed in an iterative manner within the FL fashion, namely (a) client training, (b) knowledge distillation, (c) discrimination between familiar and unfamiliar traffic packets and (d) hard-labeling and voting. During the first phase, the clients train their CNNs with private local data. In the second phase, knowledge distillation follows a teacher-student approach, where a teacher model guides the training of a student model, providing soft targets or logits. Next, a discrimination network is used from the FL server to evaluate further the predicted labels of each client's CNN. Finally, hard labeling and voting mechanisms take place in order to consider only the labels from the majority of the FL clients and proceed with the aggregation process.

2) Non-iid data in FL-based IDS: Non-iid data and data heterogeneity refers to the variation in the distribution, types, or characteristics of data across different clients. This variation poses challenges when creating FL models, as they need to generalize across diverse datasets. Towards tackling the above challenge in FL-based IDS, [17] proposes the FL-based Attention-Gated Recurrent Unit (`FedAGRU`) to address, among others, the issue of different label distribution across clients' data, thus demonstrating performance gains over conventional FL aggregation strategies. Moreover, in [18], the authors propose a peer-to-peer algorithm, namely `P2PK-SMOTE`, to train FL-driven anomaly detection models in non-iid data scenarios. The latter refers to inter and intra-imbalanced classes across the FL clients. Numerical results indicated performance gain of the proposed strategy against non-rebalancing approaches. Additionally, [19] leveraged the `Fed+` method [20] for FL-driven intrusion detection in heterogeneous networks. The clients own datasets from various types of networks, such as industrial IoT, wireless networks and wireless vehicular networks, while `Fed+` facilitates the generation of personalized local models with enhanced attack classification performance. The concept of non-iid data was supported by the assumption that the data stem from different network devices. In addition to this, in [21], the authors take `Fed+` a step further by incorporating differential privacy techniques. Next, in [22], data augmentation techniques to address class imbalance and non-iid settings are investigated. Specifically, data augmentation methods such as SMOTE, ADASYN and adversarial generative networks were invoked to support upsampling of the imbalanced client data. The evaluation results indicate a performance improvement compared with baselines that do not rely on data augmentation strategies. Finally, in [23], authors propose a clustering-enabled FL meta-learning framework to tackle class imbalance and non-iid data. Specifically, they design a data- and model-agnostic meta-sampler that adaptively balances local data sets, and thus, mitigating the data imbalance problem. Hence, the focus of this work lies mainly in dealing with class imbalance in FL-based IDS.

B. Motivation

Undoubtedly, the previous works offer valuable insights and methodologies. However, in the majority of them, the assumption of iid data across the clients is not valid within realistic FL conditions. Conventional FL strategies like FedAvg are not designed for handling non-iid data and may experience notable performance degradation or even divergence when applied in such situations [24]. Although the works [17], [18] and [16], [22], [23] successfully examine and design FL-based IDS considering non-iid data, emphasis was mainly given to the following cases: (a) class imbalance across clients datasets and/or different label distributions and (b) different number of samples per client. Therefore, the aforementioned works mainly address class and data imbalance issues. The case of non-iid features among clients' data is underrepresented, while its impact on the global model convergence remains vague. On the contrary, the authors in [19] study a broader aspect of non-iid settings by considering heterogeneous datasets across the clients. However, personalized FL methods such as Fed+ are employed, generating multiple personalized local models instead of a unified global one. Such approach prevents the generation of a single global model, that can be further distributed to third parties.

Well-known techniques in the literature that address non-iid data issues include FedProx [25], which stabilizes local training by introducing a proximal term, FedNova [26], which considers that each client may conduct a different number of local training steps, and SCAFFOLD [27], which uses control variates both at server and clients to estimate the model update direction. However, as mentioned previously, a particular example of non-iid data among clients is the case of non-iid features, which has generally received less attention in the FL-related literature. Methods addressing this issue mainly rely on layer normalization [28] and batch normalization [29] techniques. Specifically, [29] proposes FedBN, a method that incorporates batch normalization layers on local clients' model, which are not included in the aggregation step at the server side. Although FedBN has shown potential in mitigating non-iid features, it assumes that clients possess batch normalization layers and have been actively involved in the FL training. Consequently, non-participating clients that may want to access the global model are excluded, as the method cannot generate a universally applicable global model. This fact implies limitations in distributing the global model to additional entities. Moreover, the work in [30], proposed a FL/split learning method to address non-iid data in a user authentication scenario. The method involves splitting a global model trained initially on a public dataset, into two parts: a feature extractor subnetwork and a classifier head. Clients compute the mean and variance of the feature extractor based on their local datasets and send these statistics to the server. The server then generates a synthetic dataset by sampling from the aggregated client statistics. While this method proved effective, it assumes the availability of a pre-existing public dataset on the server and depends on data augmentation techniques. Finally, as per [31] experimental study, none of the existing state-of-the-art FL methods and aggregation strategies for non-iid data outperform the other ones in all cases. Therefore, exploring novel techniques to address the impact of data heterogeneity in terms of non-iid features, particularly within FL-based IDS, which is still immature in the context of the mentioned challenge, is an interesting and promising topic. To the best of our knowledge, limited attention has been given to the issue of non-iid features among clients in FL-based IDS. Notably, the works [17]–[23] do not particularly focus on this subject.

C. Contribution

In light of the aforementioned motives, in this paper, we introduce the Statistical Averaging (StatAvg) method to circumvent the challenges of non-iid features of clients in FL. Due to different feature distributions across clients, the local data normalization process may differ from client to client. It is noted here that data normalization refers to the scaling of the input data, e.g., the scaling of features through methods such as standard scaling. Inconsistencies in feature distribution can hinder or even prevent the convergence of the federated global model, as each local model is trained on a different input data distribution. To this end, StatAvg aims at producing global data statistics that can serve as a universal normalization for the local data of each client. This approach enables clients to scale their input data (features) based on this unique global scaler. To achieve this, the server is responsible for collecting the local data statistics of the clients and afterwards aggregating them properly to produce global data statistics. In this way, clients use a shared global normalization scale, based on the aggregated data statistics, to standardize their local data, helping to reduce the impact of non-iid features in their individual datasets. The contributions of our work are summarized as follows:

- The StatAvg method is proposed to alleviate the effects of non-iid feature distributions in FL. According to StatAvg, the FL clients calculate their local data statistics, specifically the mean and variance of the input feature vector, and transmit them to the server. The server aggregates the clients' local statistics to generate global statistics. We prove mathematically that the aggregated global statistics represent the true mean and variance of the combined datasets across all clients. Afterwards, the server broadcasts the global statistics to all clients, normalizing their input features based on these global statistics.
- StatAvg enables the generation of global data statistics that can be interpreted as a universal input data normalization process and applied before feeding the data into the global model. It is important to emphasize that typically, a trained model should be accompanied by the corresponding normalization technique on the input data. Otherwise, the model will be ineffective during inference, without the proper input data normalization. However,

this aspect is often overlooked in the existing literature. Therefore, `StatAvg` serves as a means to offer a global data normalization technique that can be applied to the global model by external entities that are not necessarily involved in the training procedure.

- The performance of `StatAvg` is evaluated through experiments on two open datasets for host and network intrusion detection, namely the TON-IoT and CIC-IoT-2023. Various FL aggregation strategies are used as baseline methods for comparison, including `FedAvg`, `FedLN` [28], and `FedBN` [29]. The demonstrated results showcase the prevalence of `StatAvg` over the baselines in terms of evaluation metrics such as the detection accuracy and the F1 score. Finally, some illustrative insights are provided that justify the presence of clients' non-iid features on the examined intrusion detection datasets.

D. Structure

The structure of the paper comes as follows. Section II provides preliminary information regarding FL and non-iid features. Next, section III presents and analyzes `StatAvg`. Finally, section IV focuses on the evaluation analysis of `StatAvg`, while section V concludes this paper.

II. Preliminaries of Federated Learning

A. FL System

We consider an FL environment consisting of N clients, indexed as $i \in \mathcal{N} = \{1, 2, ..., N\}$ and a server. Each client owns a dataset $\mathcal{D}_i = \{(\boldsymbol{x}_i^j, y_i^j) \in \mathbb{R}^S \times \mathbb{C}\}_{j=1}^{D_i}$, where $\boldsymbol{x}_i^j$ is the j-th input sample, $D_i = |\mathcal{D}_i|$ is the number of samples and S denotes the number of features. Here, $\mathbb{R}$ denotes the set of real numbers. Additionally, we denote $\mathbb{C}$ as the set to which the label y_i^j belongs, e.g., it could be a subset of the real numbers, a set of categorical values for classification tasks, etc. In this paper, $\mathbb{C}$ contains the labels of cyberattacks and will be described below in this work, along with the description of the datasets used in the evaluation experiments.

The overall dataset across all clients is denoted as $\mathcal{D} = \bigcup\limits_{i \in \mathcal{N}} \mathcal{D}_i$ and the size of all training data is $D = \sum_{n=i}^{N} D_i$. The loss function of client i, is defined as:

$$F_i(\boldsymbol{w}) \triangleq \frac{1}{D_i} \sum_{j=1}^{D_i} \phi\left(\boldsymbol{w}, \boldsymbol{x}_i^j, y_i^j\right), \quad \forall i \in \mathcal{N}, \tag{1}$$

where $\phi(\boldsymbol{w}, \boldsymbol{x}_i^j, y_i^j)$ captures the error of the model parameter $\boldsymbol{w} \in \mathbb{R}^K$ for the input-output pair $(\boldsymbol{x}_i^j, y_i^j)$, where K is the size of model parameters. The ultimate goal of the FL process is to obtain the global parameter $\boldsymbol{w}$, which minimizes the loss function on the whole dataset.

$$F(\boldsymbol{w}) = \sum_{n=1}^{N} n_i F_i(\boldsymbol{w}), \tag{2}$$

where $n_i = \frac{D_i}{D}$ is the proportion of data samples owned by client i relative to the entire dataset.

In a nutshell, the FL process is executed for a specified number of communication rounds. At the t-th round, the server firstly broadcasts the global model $\boldsymbol{w}^{(t)}$ to all clients. Each client i updates its local model $\boldsymbol{w}_i^{(t)}$ via a gradient-based method on the loss function F_i and uploads it to the server. Finally, the server generates the global model $\boldsymbol{w}^{(t+1)}$ by using an aggregation strategy of its preference. The aforementioned process is repeated for the selected number of rounds until the convergence of the global model is achieved.

B. Non-iid features in FL

In line with the definitions provided by [24] and [29], the presence of non-iid features across clients can be expressed through the following concepts:

- *Feature distribution skew* (covariate shift): The marginal distributions $P_i(\boldsymbol{x})$ varies across clients, even if $P_i(y|\boldsymbol{x})$ is the same for all clients.
- *Same label, different features* (concept drift): The conditional distributions $P_i(\boldsymbol{x}|y)$ may vary across clients even if $P_i(y)$ is common. As such, the same label y can have different features $\boldsymbol{x}$ for different clients.

Non-iid features can significantly degrade the performance of FL, by introducing inconsistencies in model updates across clients. Since each client is exposed to different input distributions, their local models may learn patterns that do not generalize well to other clients. This inconsistency can result in unstable training, where the global model struggles to converge in a timely manner. In extreme cases, the divergence between local models can be so severe that the global model completely fails to converge. These challenges make it difficult for the server to effectively aggregate the locally trained models into a coherent global model that performs well across all clients. Consequently, the presence of non-iid features requires specialized techniques or modifications to standard FL algorithms to ensure successful training and generalization.

III. StatAvg - Statistical Averaging

A. Description and Algorithm

Traditionally, individual FL clients normalize their local data based on their own local statistics, with the most prominent normalization technique being the z-score normalization, i.e., clients subtract the mean from each data sample of a given feature and then divide it with the standard deviation. This is equivalent to shifting the input feature distribution to have a zero mean and unit variance. Accordingly, in the testing phase, the testing dataset is scaled based on the aforementioned normalization, individually per client. In the presence of non-iid features between clients, the local normalization process may significantly differ from client to client. As a result, this variability may affect the convergence of the global FL model since each local model is trained on a different input data distribution. To tackle the issue of non-iid features across clients, our objective is to discover global statistics that clients can share without requiring access to their raw data. Typical statistical metrics include the mean and variance of the features, whereas this study investigates the impact of these particular metrics.

In the light of the previous discussion, we proceed to compute the mean and variance for each client's features. The mean value across all samples of a feature $s \in \mathcal{S}$ of client i, where $\mathcal{S}$ is the entire feature set, is given as:

$$\mu_{i,s} = \frac{1}{D_i} \sum_{j=1}^{D_i} x^j_{i,s} \tag{3}$$

and $\boldsymbol{\mu}_i = (\mu_{i,1}, \mu_{i,2}, ..., \mu_{i,S})$ is the vector with all the means of each feature. Its is worth noting that $x^j_{i,s}$ is the s-th entry of $\boldsymbol{x}^j_i$. Accordingly, the corresponding variance is calculated as:

$$\sigma^2_{i,s} = \frac{1}{D_i} \sum_{j=1}^{D_i} \left(x^j_{i,s} - \mu_{i,s} \right)^2 \tag{4}$$

and $\boldsymbol{\sigma}^2_i = (\sigma^2_{i,1}, \sigma^2_{i,2}, ..., \sigma^2_{i,S})$. Hereinafter, with the term *local statistics* of client i, we refer to the tuple $\{\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i\}$. The StatAvg strategy aims at obtaining the global statistics $\{\boldsymbol{\mu}_{\mathrm{G}}, \boldsymbol{\sigma}^2_{\mathrm{G}}\}$ of the overall dataset $\mathcal{D}$ by aggregating the local statistics $\{\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i\}_{i\in\mathcal{N}}$. In this manner, all clients can normalize their data based on global statistics, which guarantees a common normalization/scaling of the input data. The detailed process of StatAvg is described in Algorithm 1.

```
Algorithm 1 StatAvg
Input: $\mathcal{N}$, $\{\mathcal{D}_i\}_{i\in\mathcal{N}}, \boldsymbol{w}^{(1)}$
Output: $\boldsymbol{w}$, $\{\boldsymbol{\mu}_{\mathrm{G}}, \boldsymbol{\sigma}^2_{\mathrm{G}}\}$
 1: for $t = 0, 1, 2, ...$ do
 2:     if $t = 0$ then
 3:         for each client $i \in \mathcal{N}$ do
 4:             calculate $\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i$ according to (3), (4)
 5:             send $\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i$ to the server
 6:         server calculates the global statistics as:
                $\boldsymbol{\mu}_{\mathrm{G}} = \sum_{i\in\mathcal{N}} n_i \boldsymbol{\mu}_i$,
            $\boldsymbol{\sigma}^2_{\mathrm{G}} = \sum_{i\in\mathcal{N}} n_i \left( \boldsymbol{\sigma}^2_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{\mathrm{G}})^2 \right)$
 7:         server sends $\{\boldsymbol{\mu}_{\mathrm{G}}, \boldsymbol{\sigma}^2_{\mathrm{G}}\}$ to all clients
 8:         for each client $i \in \mathcal{N}$ do
 9:             normalize input features as:
                $\tilde{x}^j_{i,s} = \frac{x^j_{i,s} - \mu_{\mathrm{G},s}}{\sigma_{\mathrm{G},s}}, \quad \forall j \in \{1, ..., D_i\}, \ \forall s \in \mathcal{S}$
10:             $\tilde{\mathcal{D}}_i = \{(\tilde{\boldsymbol{x}}^j_i, y^j_i)\}^{D_i}_{j=1}$
11:     else                                        ▷ standard FL procedure
12:         server sends $\boldsymbol{w}^{(t)}$ to all clients $i \in \mathcal{N}$
13:         for each client $i \in \mathcal{N}$ do
14:             $\boldsymbol{w}^{(t)}_i = \boldsymbol{w}^{(t)}_i - \eta \nabla F_i \left( \boldsymbol{w}^{(t)}_i, \boldsymbol{\xi}^{(t)}_i \right), \quad \boldsymbol{\xi}^{(t)}_i \subseteq \tilde{\mathcal{D}}_i$
15:             send $\boldsymbol{w}^{(t)}_i$ to the server
16:         $\boldsymbol{w}^{(t+1)} = \sum_{i\in\mathcal{N}} n_i \boldsymbol{w}^{(t)}_i$
17: $\boldsymbol{w} = \boldsymbol{w}^{(t)}$
```

As can be seen, the StatAvg strategy occurs solely during the first round ($t = 0$), prior to the actual FL training. Firstly, in steps 2 - 4, each client calculates its local statistics and sends them to the server. Following that, in steps 5 - 6, the server calculates the global statistics based on the received local statistics and broadcasts them back to the clients. It is worth mentioning that the operations in step 5 are carried out element-wise. The rationale behind the aggregation technique used to obtain $\boldsymbol{\mu}_{\mathrm{G}}$ and $\boldsymbol{\sigma}^2_{\mathrm{G}}$ is explained later in this work. Afterwards, in steps 7 - 8, the clients normalize their input features based on the global statistics by utilising conventional z-score normalization. It should be highlighted that the communication overhead for exchanging the local and global statistics between the clients and the server is negligible since it takes place solely during the first round. Additionally, the size of the local statistics tuple is negligible compared to the size of the local model, because the number of features is typically much smaller than the number of model parameters (weights) used during training, i.e., $2S \ll K$. At step 10 and afterwards, a conventional FL process follows, e.g., FedAvg, that will ultimately generate the global FL model. However, the selection of the aggregation strategy is not limited to FedAvg and can vary according to the particularities of the underlying FL task. Note also that during the client local update in step 13, η is the learning rate and $\boldsymbol{\xi}^{(t)}_i \subseteq \tilde{\mathcal{D}}_i$ is a randomly sampled mini-batch from the normalized local dataset $\tilde{\mathcal{D}}_i$. Finally, if the local dataset $\mathcal{D}_i$ changes dynamically in each round (it can be denoted as $\mathcal{D}^{(t)}_i$), applying StatAvg in such case is straightforward. This can be done by computing the local statistics in each round and constructing the normalized dataset $\tilde{\mathcal{D}}^{(t)}_i$, based on the global statistics of the given round.

It should be again clarified that StatAvg focuses on the aggregation of statistical metrics rather than local models $\boldsymbol{w}^{(t)}_i$, facilitating its integration with any model aggregation strategy. Fig. 1 provides an illustration of StatAvg's implementation. Finally, we stress that through StatAvg, a universal input data normalization technique is provided. This is a crucial remark since a trained model should be paired with the appropriate data normalization technique (also known as *scaler*) to render it effective during inference.

In the continue, we will show that $\boldsymbol{\mu}_{\mathrm{G}}$ and $\boldsymbol{\sigma}^2_{\mathrm{G}}$ are the mean and variance of the overall dataset $\mathcal{D}$. First, we assume that $\mathcal{D}_i \cap \mathcal{D}_k = \emptyset$, $\forall i, k \in \mathcal{N}$, $i \neq k$. This implies that all local datasets are pairwise disjoint. The assumption is reasonable, considering that each dataset originates from a distinct client, thus making it highly unlikely - if not impossible - for identical samples to appear across different local datasets. To this end, we proceed to formulate the following proposition.

Proposition 1: Let $\boldsymbol{x}_{i,s} \in \mathbb{R}^{D_i}$ be the vector containing the s-th feature across all samples of $\mathcal{D}_i$. Also, let $\boldsymbol{z}_s = (\boldsymbol{x}_{1,s}, ..., \boldsymbol{x}_{N,s})$ be the concatenation of all clients vectors, with $\boldsymbol{z}_s \in \mathbb{R}^D$. The mean and variance of $\boldsymbol{z}_s$ are given as

$$\begin{aligned} \mu_{\mathrm{G},s} &= \sum_{i\in\mathcal{N}} n_i \mu_{i,s} \\ \sigma^2_{\mathrm{G},s} &= \sum_{i\in\mathcal{N}} n_i \left(\sigma^2_{i,s} + (\mu_{i,s} - \mu_{\mathrm{G},s})^2 \right). \end{aligned} \tag{5}$$

Proof:

First, the notation of s is dropped for the simplicity of presentation. It is straightforward to compute the mean of $\boldsymbol{z}$ as:

$$
\begin{aligned}
\mu_{\mathrm{G}} &= \frac{1}{D}\sum_{l=1}^{D} z_l = \frac{1}{D}\sum_{i=1}^{N}\sum_{j=1}^{D_i} x_i^j = \frac{1}{D}\sum_{i=1}^{N} D_i \sum_{j=1}^{D_i} \frac{x_i^j}{D_i} \\
&= \sum_{i=1}^{N} n_i \mu_i .
\end{aligned} \tag{6}
$$

Before examining σ_{G}^2, first, it is noted that for the local variances, it holds:

$$
\sigma_i^2 n_i = \sum_{j=1}^{D_i} (x_i^j - \mu_i)^2, \quad \forall i \in \mathcal{N}. \tag{7}
$$

Similarly, for $\boldsymbol{z}$ we get

$$
\begin{aligned}
\sigma_{\mathrm{G}}^2 D &= \sum_{l=1}^{D} (z_l - \mu_{\mathrm{G}})^2 = \sum_{i=1}^{N}\sum_{j=1}^{D_i} (x_i^j - \mu_{\mathrm{G}})^2 \\
&= \sum_{i=1}^{N}\sum_{j=1}^{D_i} \left((x_i^j)^2 - 2x_i^j \mu_{\mathrm{G}} + \mu_{\mathrm{G}}^2 \right) .
\end{aligned} \tag{8}
$$

The inner sum in the last term of (8) can be expanded by adding and subtracting μ_i^2, as:

$$
\begin{aligned}
&\sum_{j=1}^{D_i} \left((x_i^j)^2 - 2x_i^j \mu_{\mathrm{G}} + \mu_{\mathrm{G}}^2 + \mu_i^2 - \mu_i^2 \right) \\
&= \sum_{j=1}^{D_i} \left((x_i^j - \mu_i)^2 + 2x_i^j (\mu_i - \mu_{\mathrm{G}}) + \mu_{\mathrm{G}}^2 - \mu_i^2 \right) \\
&= \sum_{j=1}^{D_i} (x_i^j - \mu_i)^2 + 2D_i\mu_i(\mu_i - \mu_{\mathrm{G}}) + D_i\mu_{\mathrm{G}}^2 - D_i\mu_i^2 \\
&= D_i\sigma_i^2 + D_i\mu_{\mathrm{G}}^2 - 2D_i\mu_i\mu_{\mathrm{G}} + D_i\mu_i^2 \\
&= D_i\sigma_i^2 + D_i(\mu_i - \mu_{\mathrm{G}})^2 .
\end{aligned} \tag{9}
$$

By combining (9) with (8) we conclude to

$$
\sigma_{\mathrm{G}}^2 = \sum_{i \in \mathcal{N}} n_i \left(\sigma_i^2 + (\mu_i - \mu_{\mathrm{G}})^2 \right), \tag{10}
$$

which completes the proof. ■

Proposition 1 provides a way to obtain the global mean and variance across the whole dataset $\mathcal{D}$ for a given feature s. The proof can be easily generalized $\forall s \in \mathcal{S}$, which gives rise to the vector representation of the global mean and variance for each feature, i.e., $\boldsymbol{\mu}_{\mathrm{G}}$ and $\boldsymbol{\sigma}_{\mathrm{G}}^2$, respectively. This result is used in step 5 of Algorithm 1 to derive the global mean and variance.

B. Differential Privacy Extension

It is clarified that the local statistics being shared with the server are high-level, aggregated summaries of the data and do not reveal individual data points or sensitive attributes. Therefore, these statistics lack sufficient granularity to reconstruct the underlying dataset or any individual client's private information. However, to further enhance privacy, differential privacy (DP) strategies could be easily integrated into the proposed method during the transmission of the local statistics [32]. Specifically, by adding a controlled amount of random noise to the local statistics, DP ensures that individual client contributions cannot be easily inferred by the server.

According to DP principles, instead of directly sending the local statistics $\{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}$ to the server, the clients perturbs them and send a distorted version. Specifically, to ensure (ϵ, δ)-DP [32], for a given feature $s \in \mathcal{S}$, client i adds Gaussian noise to $\mu_{i,s}$ and $\sigma_{i,s}^2$ as follows:

$$
\begin{aligned}
\tilde{\mu}_{i,s} &= \mu_{i,s} + \text{Gaussian}\left(0, \zeta^2_{\mu_{i,s}}\right), \\
\tilde{\sigma}^2_{i,s} &= \sigma^2_{i,s} + \text{Gaussian}\left(0, \zeta^2_{\sigma^2_{i,s}}\right),
\end{aligned} \tag{11}
$$

where the variance of the noise for $\mu_{i,s}$ and $\sigma_{i,s}^2$ are described, respectively as

$$
\begin{aligned}
\zeta^2_{\mu_{i,s}} &= \frac{2\ln(1.25/\delta)}{\epsilon^2} \Delta^2_{\mu_{i,s}}, \\
\zeta^2_{\sigma^2_{i,s}} &= \frac{2\ln(1.25/\delta)}{\epsilon^2} \Delta^2_{\sigma^2_{i,s}}
\end{aligned} \tag{12}
$$

and the sensitivities $\Delta^2_{\mu_{i,s}}$, $\Delta^2_{\sigma^2_{i,s}}$ of the mean and variance functions are given by

$$
\begin{aligned}
\Delta^2_{\mu_{i,s}} &= \frac{\max_j\{x_{i,s}^j\} - \min_j\{x_{i,s}^j\}}{D_i}, \\
\Delta^2_{\sigma^2_{i,s}} &= \frac{\left(\max_j\{x_{i,s}^j\} - \min_j\{x_{i,s}^j\}\right)^2}{D_i},
\end{aligned} \tag{13}
$$

accordingly. By applying the above process for each feature $s \in \mathcal{S}$ independently, each client i creates the peturbated local statistics $\{\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\sigma}}_i^2\}$ and sends them to the server, which then proceeds with the conventional aggregation process.

IV. Evaluation Analysis

This section presents experiments conducted on different datasets to detect intrusions in a federated setting. The effectiveness of the proposed strategy `StatAvg` is evaluated by comparing it with various baseline methods.

A. Evaluation Datasets

The experiments were conducted on the following well-known public datasets.

TON-IoT Dataset: Among others, the TON-IoT Dataset [33] includes operating system data of Ubuntu versions 14 and 18, which is adopted in our work. More specifically, it includes audit traces documenting memory activities within the operating system. The dataset is suitable for training and designing host-based IDS. Also, the dataset is composed of data stemming from various physical or virtual devices belonging to the edge and cloud layers. The description of the selected features is provided in Table II. Furthermore, the attacks on the host system that serve as the labels of the dataset are "dDoS", "DoS", "Injection", "Password", "Mitm", while also a class named "Normal" is included, indicating the normal behaviour of the host system. More details regarding the dataset can be found in [33] and [34].

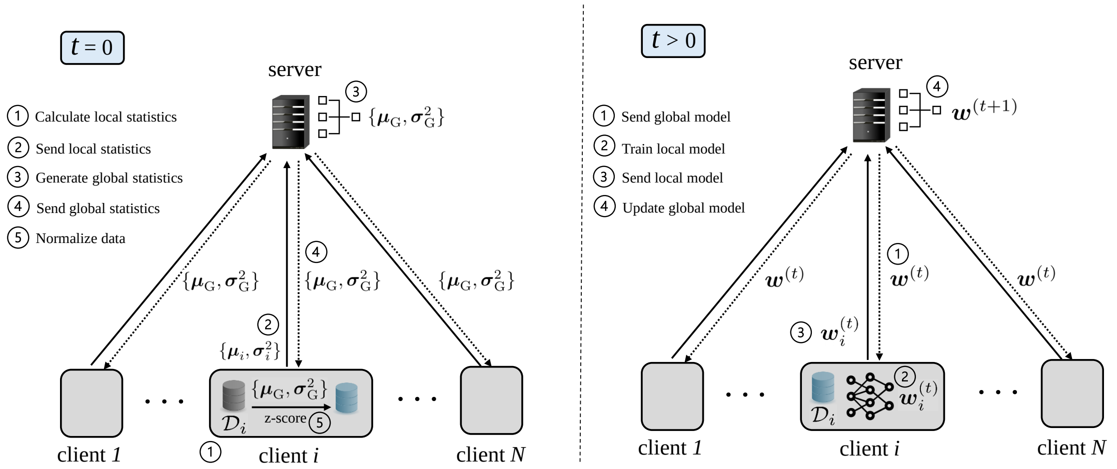


Fig. 1. Visual representation of StatAvg design and implementation.

TABLE I
LIST OF NOTATIONS

Parameter	Description	Parameter	Description
$\mathcal{N}$	Set of clients	$P(\cdot)$	Probability density function
i	Indexing of clients	$\boldsymbol{\mu}_i$	The mean values vector of the features for the i-th client
$\mathcal{D}_i$	Dataset of client i	$\boldsymbol{\sigma}_i^2$	The variances vector of the features for the i-th client
D_i	Dataset size of client i	$\boldsymbol{\mu}_{\mathrm{G}}$	Global mean values vector
$\boldsymbol{x}_i^j$	j-th input sample of client i	$\boldsymbol{\sigma}_{\mathrm{G}}^2$	Global variances vector
S	Number of features	$\boldsymbol{z}$	Concatenated vector
s	Index of features	z_l	The l-th element of the vector $\boldsymbol{z}$
y_i^j	j-th label of client i	t	FL round index
$\mathcal{D}$	Overall dataset of all clients	n_i	Proportion of data samples owned by client i
D	Size of the overall dataset	$\boldsymbol{w}^{(t)}$	Global model at round t
N	Total number of clients	$\boldsymbol{w}_i^{(t)}$	Local model of i-th client at round t
K	Size of model parameters	$\mathbb{C}$	Set of classes
$\tilde{\mathcal{D}}_i$	Normalized dataset of client i	$\boldsymbol{\xi}_i^{(t)}$	Random mini-batch of client i at round t

CIC-IoT-2023 Dataset: The CIC-IoT-2023 Dataset [35] is a realistic IoT attack dataset, using an extensive topology composed of multiple IoT devices designated as either attackers or targets. The dataset entails 48 features that are characterized by metrics such as packet flow statistics, employed application layer protocols, TCP flags, etc. As we do not explicitly describe all features for brevity, additional information can be found in [35]. Furthermore, the dataset categorizes attacks into eight classes, namely “Brute force”, “dDoS”, “DoS”, “Mirai”, “Recon”, “Spoofing”, “Web-based”, and “Normal”.

B. Baseline Aggregation Methods

To evaluate the performance of `StatAvg`, we use the following baseline aggregation strategies:
`FedAvg`: It is the de facto approach for FL [36]. Clients perform local model updates and the server executes the aggregation of the local models to generate the global model.

`FedLN`: The layer normalization is included in the local models for mitigating the effects of non-iid features [28]. `FedLN` performs local updates and averages local models similarly to `FedAvg`.

TABLE II
TON-IOT DATASET: FEATURE DESCRIPTION

Feature name	Description
MINFLT	The number of page faults issued by this process that have been solved by reclaiming the requested memory page from the free list of pages
MAJFLT	The number of page faults issued by this process that have been solved by creating/loading the requested memory page
VSTEXT	The virtual memory size used by the shared text of this process
VSIZE	The total virtual memory usage consumed by this process
RSIZE	The total resident memory usage consumed by this process
VGROW	The amount of virtual memory that the process has grown during the last interval
RGROW	The amount of resident memory that the process has grown during the last interval
MEM	Memory occupation percentage

`FedBN`: Employs local batch normalization (BN) to the local models prior to averaging them towards alleviating feature shift. Nonetheless, `FedBN` assumes that local models have BN layers and omits their parameters from the aggregation step at the side of the server [29].

It is worth noting that `FedLN` and `FedBN` are specially tailored to address the issue of non-iid features, justifying their selection. As follows, in `FedAvg`, `FedLN` and `FedBN`, the normalization of the local training data is performed based on the local client statistics, aligning with the conventional FL approach. Also, the testing data undergo scaling in accordance with the respective local normalization for each client individually. Finally, it is noted that the proposed technique `StatAvg` utilizes `FedAvg` at step 9 of Algorithm 1 as the default model aggregation strategy.

C. Experimental Setup

The following settings apply to all experiments unless specified otherwise. The number of clients has been set as $N = 5$ and $N = 10$, while all clients are considered to participate in every FL round. Also, the number of FL rounds is set to 50 and 80, for $N = 5$ and $N = 10$, respectively. Each client receives an equal proportion $n_i = \frac{1}{N}$ of the original dataset $\mathcal{D}$. Moreover, the division is conducted through stratification based on the labels of the original dataset, aiming to approximate a common $P_i(y)$ across all clients. This implies that clients share a common label distribution. Following that, each client splits its local dataset into training and testing subsets, with a ratio of 4 to 1. Due to the significant class imbalance in the datasets, each client generates synthetic instances from the minority classes in the training set by using SMOTE [37]. It is noted that SMOTE is applied independently on each client, after the splitting of the overall dataset.

The local model of each client is a neural network consisting of 3 Fully Connected (FC) hidden layers with 128 neurons and ReLU activation, denoted as (FC(128), ReLU), followed by a softmax activation on the output layer. In the case of the baselines FedLN and FedBN, layer normalization (LN) and batch normalization (BN) layers are incorporated into the local models, resulting in each layer being structured as (FC(128), ReLU, LN) and (FC(128), BN, ReLU), respectively. For the local training updates, the Adam optimizer is adopted [38]. Finally, additional settings are summarized in Table III [1]

D. Evaluation Results

Regarding the performance evaluation, we use common evaluation metrics such as the confusion matrix, accuracy, and F1 score. Given a specific attack/class, the confusion matrix includes the following standard metrics: the True Positive (TP) represents instances where the model correctly identifies a sample as belonging to a specific attack type. True Negative (TN) counts instances where the model accurately identifies a sample as not belonging to a specific attack type when it truly does not. False Positive (FP) denotes instances where the sample is predicted as of a certain attack, but actually, the sample does not belong to that attack type. False Negative (FN) is the number of instances for which the model fails to predict a sample as a specific attack type, even though the sample actually belongs to that attack. Next, the accuracy and F1 score are defined as:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{14}$$

and

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{15}$$

respectively.

The evaluation metrics showcased in the results have been averaged across all classes due to the multi-class nature of the problems we are addressing. Finally, it is noted the evaluation was performed using clients' testing sets, and the demonstrated results were also averaged across all clients.

First, the evolution of testing accuracy throughout the FL rounds is evaluated, for both $N = 5$ and $N = 10$ clients. In Fig. 2, 3, and Fig. 4, 5, the `StatAvg` strategy is compared with the selected baselines on the TON-IoT and CIC-IoT-2023 datasets, respectively. It is evident that `StatAvg` significantly outperforms the baseline strategies across both datasets in terms of accuracy. Moreover, the convergence curve of `StatAvg` is more stable compared to that of the baseline methods, which display higher variance. The exhibited performance gain lies in the fact that `StatAvg` utilizes global statistics to normalize the clients' features. Although FedLN and FedBN have been designed to minimize the effects of non-IID features between clients, it is discernible that they struggle to address this issue in certain datasets. The variations in local client statistics, and consequently, the diverse local normalization utilized, appear to degrade the performance of FL. Additionally, it is observed that `Statavg` consistently outperforms the baseline methods in both client settings, demonstrating its robustness to client scaling.

Moreover, in Table IV and Table V, some evaluation metrics for the case of TON-IoT and CIC-IoT-2023 datasets are demonstrated, respectively. The considered metrics showcase the performance of the best models encountered during the FL training for each strategy. It can be observed that `StatAvg` has superior performance against the baseline strategies. Specifically, in the case of the TON-IoT dataset, `StatAvg` demonstrates a notable improvement, for both settings of clients, of over 17% and 16% in accuracy and F1 score, respectively, compared to the second-best strategy FedLN. Also, when considering the CIC-IoT-2023 dataset, the corresponding increase is over 4% and 2% for accuracy and F1 score. The detailed confusion matrices of the `StatAvg` metho, when considering $N = 5$ clients, are presented in Fig. 6 and Fig. 7, for the TON-IoT and CIC-IoT-2023 datasets, respectively. In both datasets, it is evident that some classes are easier to classify, e.g., "DDoS", "DoS", "Mirai", and "Normal". This can be attributed to the large number of samples that these classes usually have (e.g., "DoS" and "Normal" are majority classes in both datasets), as well as their more recognizable traffic patterns. On the other hand, certain classes are often misclassified, e.g., "Brute Force", "Recon", "Spoofing" in Fig. 7, likely due to the similarity in their underlying traffic patterns [35].

To shed light on the concept of non-iid features, we present some illustrative examples derived from the examined datasets. First, we take a deeper look into the training samples of the

[1] In the spirit of reproducible research, the code used for the numerical results is available at: https://flower.ai/docs/baselines/statavg.html

TABLE III
EXPERIMENTAL SETTINGS

Datasets	TON-IoT	CIC-IoT-2023
training samples per client	19616 ($N = 5$) / 9808 ($N = 10$)	64050 ($N = 5$) / 32025 ($N = 10$)
training samples per client (SMOTE upsampling)	84000 ($N = 5$) / 42000 ($N = 10$)	373176 ($N = 5$) / 186584 ($N = 10$)
local training epochs	2	1
batch size	512 ($N = 5$) / 256 ($N = 10$)	1024 ($N = 5$) / 512 ($N = 10$)
learning rate	0.002	0.01

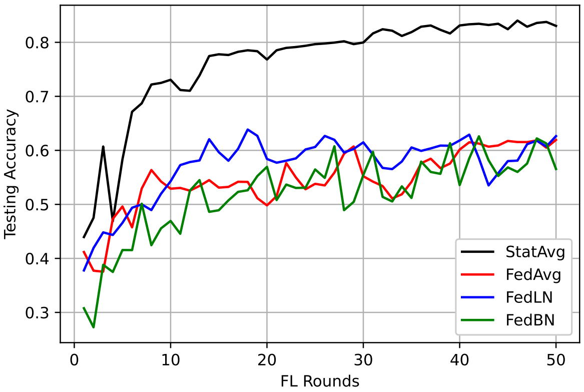


Fig. 2. Testing accuracy on TON-IoT dataset ($N = 5$ clients).

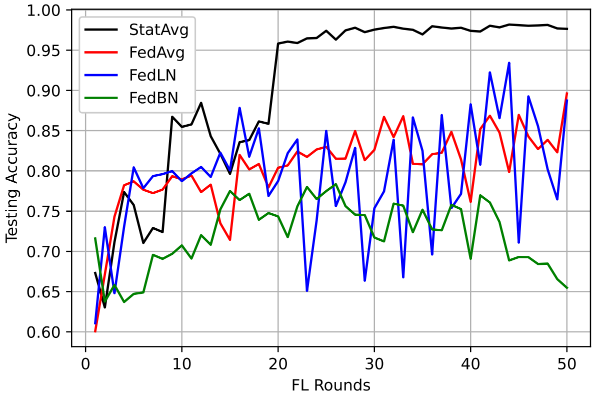


Fig. 4. Testing accuracy on CIC-IoT-2023 dataset ($N = 5$ clients).

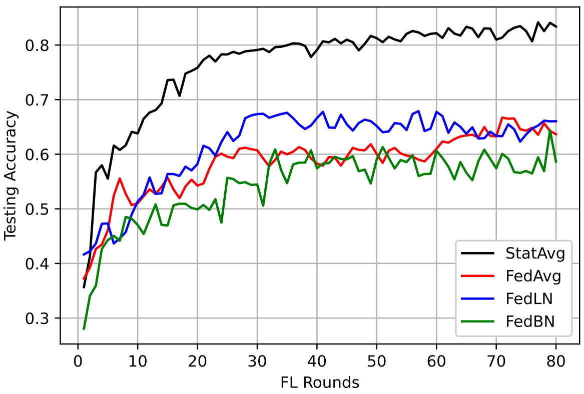


Fig. 3. Testing accuracy on TON-IoT dataset ($N = 10$ clients).

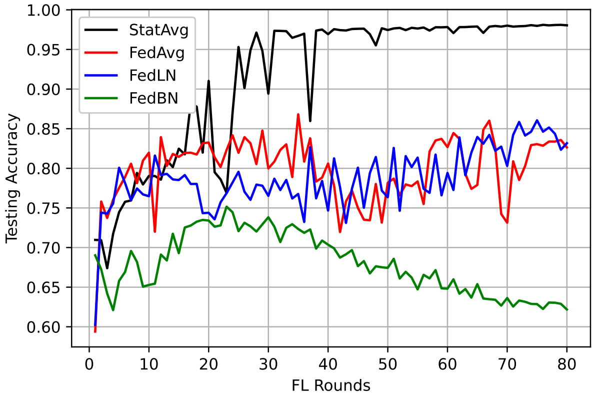


Fig. 5. Testing accuracy on CIC-IoT-2023 dataset ($N = 10$ clients).

CIC-IoT-2023 dataset, focusing specifically on those labelled with the attack category $y =$ "Web-based". Fig. 8 illustrates the distribution of the feature "Flow Duration" for the clients $i = \{1, 2\}$, formally written as $P_i(x_{i,s}|y =$ "Web-based") where $s =$ "Flow Duration". It can be observed from Fig. 8 (a) that the distributions of the clients differ. Nevertheless, it remains uncertain whether this disparity in distributions is inherent or if it is related to the limited number of samples within the selected class. It is worth noting that the "Web-based" class is indeed a minority class. From Fig. 8 (b), it is evident that the difference in distributions persists after upsampling the dataset via SMOTE. This example shows that even if $P_i(y)$ is approximately the same for all clients, as previously explained in the experimental setup, the conditional distributions $P_i(\boldsymbol{x}|y)$ can still differ. This phenomenon is related to the concept of *Same label, different features*, discussed in Section II. Another example that highlights the differences in the distributions of features is presented in Table VI. Here, statistical metrics for selected features from the TON-IoT dataset have been calculated. It can be observed that the feature "VSIZE" demonstrates consistent mean and variance across clients, while the feature "MINFLT" displays high variations in the statistical metrics. This example highlights the statistical differences that some features may exhibit among clients,

TABLE IV
EVALUATION METRICS ON TON-IOT DATASET

$N = 5$ clients				
Strategy	ACC	TPR	FPR	F1
`StatAvg`	**83.93%**	**69.26%**	**3.13%**	**62.36%**
`FedAvg`	63.68%	48.7%	8.22%	38.30%
`FedLN`	64.29%	48.9%	7.69%	40.73%
`FedBN`	62.60%	46.85%	8.66%	36.99%

$N = 10$ clients				
Strategy	ACC	TPR	FPR	F1
`StatAvg`	**83.38%**	**69.66%**	**3.26%**	**61.22%**
`FedAvg`	65.73%	52.19%	7.49%	43.55%
`FedLN`	66.07%	53.27%	7.41%	44.58%
`FedBN`	64.38%	46.29%	8.14%	38.89%

TABLE V
EVALUATION METRICS ON CIC-IOT-2023 DATASET

$N = 5$ clients				
Strategy	ACC	TPR	FPR	F1
`StatAvg`	**97.64%**	**76.01%**	**0.33%**	**75.63%**
`FedAvg`	89.71%	70.16%	3.34%	69.82%
`FedLN`	93.42%	73.41%	1.58%	73.59%
`FedBN`	78.34%	69.33%	4.10%	66.32%

$N = 10$ clients				
Strategy	ACC	TPR	FPR	F1
`StatAvg`	**98.11%**	**74.18%**	**0.28%**	**75.39%**
`FedAvg`	86.81%	68.57%	3.84%	65.88%
`FedLN`	86.06%	70.78%	2.71%	70.27%
`FedBN`	74.73%	66.40%	4.44%	63.50%

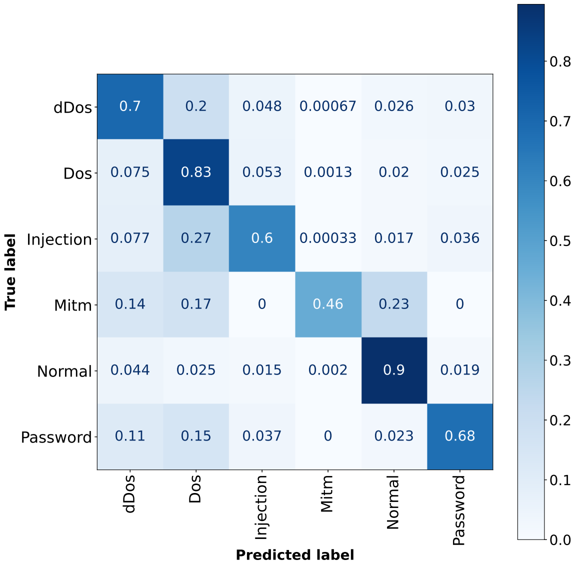


Fig. 6. Confusion matrix of StatAvg on TON-IoT dataset.

which in turn influences the local normalization of the features and potentially hinders the FL stability and convergence.

V. CONCLUSIONS

This paper proposes the `StatAvg` technique for mitigating the impact of non-iid features among clients in FL settings. The key aspect of `StatAvg` is to produce global data statistics based on the local data statistics of FL clients. The generation of global statistics, which is carried out by the server, gives rise to a universal data normalization technique that is performed by all clients. Particular attention is given to FL-based IDS, which is the focus of the experiments that were conducted. The results corroborate the effectiveness of `StatAvg` in providing robust FL convergence and classifying cyber-attacks compared to various baseline FL schemes. Moreover, valuable insights are offered within the scope of non-iid features among clients for the selected intrusion detection datasets. Finally, as `StatAvg` precedes the actual FL procedure, it can be combined with any FL aggregation strategy, a topic which is left for future investigation. Moreover, the applicability of `StatAvg` is not limited solely to FL-based IDS, as its efficacy

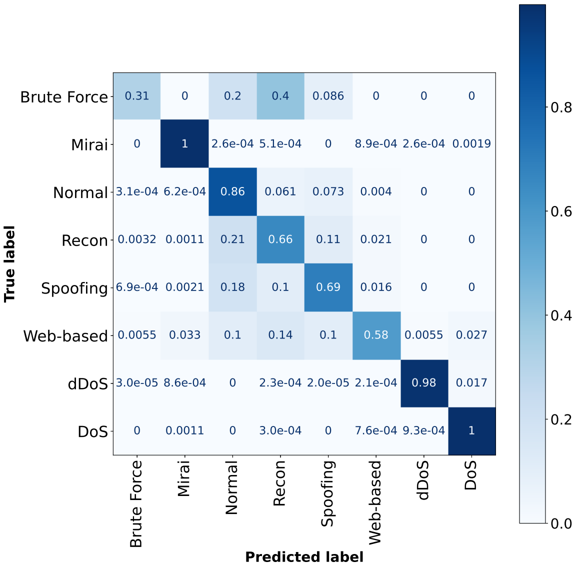
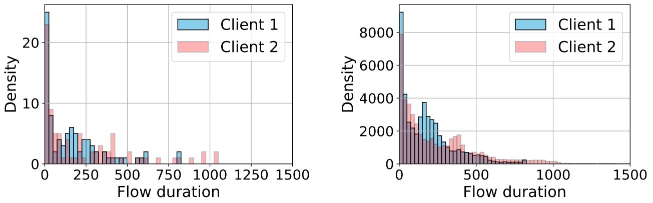


Fig. 7. Confusion matrix of StatAvg on CIC-IoT-2023 dataset.



(a) Unprocessed dataset.

(b) Upsampled (SMOTE) dataset.

Fig. 8. Distribution of the feature "Flow Duration", given the attack label "Web-based", on CIC-IoT-2023 dataset.

TABLE VI
STATISTICAL METRICS OF CLIENTS' FEATURES ON TON-IOT DATASET

Feature name	MINFLT		VSIZE	
	Mean	Variance	Mean	Variance
Client 1	694.1	$5.8 \cdot 10^6$	8621.3	$1.32 \cdot 10^8$
Client 2	694.8	$3.5 \cdot 10^6$	8663.3	$1.33 \cdot 10^8$
Client 3	735.7	$5.8 \cdot 10^7$	8364.9	$1.26 \cdot 10^8$
Client 4	691.3	$3.9 \cdot 10^6$	8521.9	$1.31 \cdot 10^8$
Client 5	769.8	$1.8 \cdot 10^8$	8519.1	$1.3 \cdot 10^8$

may encompass any FL application associated with non-iid features among clients.

References

[1] X. Deng, B. Chen, X. Chen, X. Pei, S. Wan, and S. K. Goudos, "Trusted edge computing system based on intelligent risk detection for smart iot," *IEEE Transactions on Industrial Informatics*, 2023.

[2] M. Taddeo, T. McCutcheon, and L. Floridi, "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Machine Intelligence*, vol. 1, no. 12, pp. 557–560, 2019.

[3] X. Deng, L. Wang, J. Gui, P. Jiang, X. Chen, F. Zeng, and S. Wan, "A review of 6g autonomous intelligent transportation systems: Mechanisms, applications and challenges," *Journal of Systems Architecture*, p. 102929, 2023.

[4] P. Radoglou-Grammatikis, P. Sarigiannidis, P. Diamantoulakis, T. Lagkas, T. Saoulidis, E. Fountoukidis, and G. Karagiannidis, "Strategic honeypot deployment in ultra-dense beyond 5g networks: A reinforcement learning approach," *IEEE Transactions on Emerging Topics in Computing*, 2022.

[5] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, T. Lagkas, G. Fragulis, and A. Sarigiannidis, "A self-learning approach for detecting intrusions in healthcare systems," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[7] Y. Lin, Z. Gao, H. Du, J. Kang, D. Niyato, Q. Wang, J. Ruan, and S. Wan, "Drl-based adaptive sharding for blockchain-based federated learning," *IEEE Transactions on Communications*, 2023.

[8] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, vol. 195, pp. 346–361, 2022.

[9] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabe, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, 2022.

[10] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8229–8249, 2022.

[11] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on iot intrusion detection: Federated learning, game theory, social psychology, and explainable ai as future directions," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4059–4092, 2022.

[12] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[13] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2020.

[14] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An ensemble multi-view federated learning intrusion detection for iot," *IEEE Access*, vol. 9, pp. 117 734–117 745, 2021.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[16] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semi-supervised federated learning based intrusion detection method for internet of things," *IEEE Internet of Things Journal*, 2022.

[17] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, vol. 8, pp. 217 463–217 472, 2020.

[18] H. Wang, L. Muñoz-González, D. Eklund, and S. Raza, "Non-iid data re-balancing at iot edge with peer-to-peer federated learning for anomaly detection," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021, pp. 153–163.

[19] S. I. Popoola, G. Gui, B. Adebisi, M. Hammoudeh, and H. Gacanin, "Federated deep learning for collaborative intrusion detection in heterogeneous networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–6.

[20] A. Kundu, P. Yu, L. Wynter, and S. H. Lim, "Robustness and personalization in federated learning: A unified approach via regularization," in *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*. IEEE, 2022, pp. 1–11.

[21] P. Ruzafa-Alcázar, P. Fernández-Saura, E. Mármol-Campos, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal-Bernabe, and A. F. Skarmeta, "Intrusion detection based on privacy-preserving federated learning for the industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1145–1154, 2021.

[22] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, "Enhancing iot anomaly detection performance for federated learning," *Digital Communications and Networks*, vol. 8, no. 3, pp. 314–323, 2022.

[23] W. Han, J. Peng, J. Yu, J. Kang, J. Lu, and D. Niyato, "Heterogeneous data-aware federated learning for intrusion detection systems via metasampling in artificial intelligence of things," *IEEE Internet of Things Journal*, 2023.

[24] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[26] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[27] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[28] Z. Du, J. Sun, A. Li, P.-Y. Chen, J. Zhang, H. H. Li, and Y. Chen, "Rethinking normalization methods in federated learning," in *Proceedings of the 3rd International Workshop on Distributed Machine Learning*. Rome, Italy: Association for Computing Machinery, 2022, pp. 16–22.

[29] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[30] P. Oza and V. M. Patel, "Federated learning-based active authentication on mobile devices," in *2021 IEEE international joint conference on biometrics (IJCB)*. IEEE, 2021, pp. 1–8.

[31] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 965–978.

[32] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[33] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[34] N. Moustafa, M. Ahmed, and S. Ahmed, "Data analytics-enabled intrusion detection: Evaluations of ton_iot linux datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. Guangzhou, China: IEEE, 2020, pp. 727–735.

[35] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Dr. Pavlos Bouzinis received the Diploma (five years) and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2019 and 2023, respectively, where he was a member of the Wireless Communications and Information Processing Group. Currently, he works as a researcher at MetaMind Innovations P.C. His main research interests include machine learning, optimization, and intrusion detection systems. He has served as a reviewer for several scientific journals and was an exemplary reviewer of IEEE WIRELESS COMMUNICATIONS LETTERS, in 2021 (top 3% of reviewers).

Dr. Panagiotis Radoglou-Grammatikis received Diploma (five years) and PhD from the Dept. of Electrical and Computer Engineering, University of Western Macedonia, Greece, in 2016 and 2023, respectively. His main research interests focus on AI-driven cybersecurity, intrusion detection and security games. He has published more than 50 research papers in international scientific journals, conferences and book chapters, while he has received five best paper awards. He was included in Stanford University's list (shared by Elsevier) of the Top 2% of Scientists in the World for 2021 and 2022. Currently, he is working as a research director at K3Y Ltd, while he is also a postdoc researcher at the ITHACA Lab of the University of Western Macedonia and co-founder of MetaMind Innovations P.C. He is involved in several national and international projects. Finally, he is a member of IEEE, ACM and the Technical Chamber of Greece.

Ioannis Makris received his BSc in Computer Science with specialization in Artificial Intelligence and Software Engineering from the Aristotle University of Thessaloniki (AUTh) and his MSc in Business Analytics from the University of Edinburgh. Furthermore, he is a Project Management Professional (PMP) by the Project Management Institute (PMI). His interests include privacy-preserving AI techniques, interpretable machine learning, and security. He is currently working as a Network and Security Engineer/Researcher.

Dr. Thomas Lagkas is Assistant Professor at the Department of Computer Science of the Democritus University of Thrace and Director of the Laboratory of Industrial and Educational Embedded Systems. He graduated with honours from the Department of Informatics, Aristotle University of Thessaloniki and awarded PhD on Wireless Networks. He also completed MBA studies at the Hellenic Open University and received a postgraduate certificate on Teaching and Learning from The University of Sheffield. He has been scholar of the Aristotle University Research Committee and postdoctoral scholar of the National Scholarships Institute of Greece. His research interests are in the areas of IoT communications with numerous highly cited publications. Dr. Lagkas is an IEEE Senior Member, Fellow of the Higher Education Academy in the UK, and member of the Editorial Board of reputable scientific journals. Moreover, he actively participates in several EU-funded research projects.

Prof. Vasileios Argyriou received the B.Sc. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2001, and the M.Sc. and Ph.D. degrees in electrical engineering working on registration from the University of Surrey, in 2003 and 2006, respectively. From 2001 to 2002, he held a research position with Aristotle University, with a focus on image and video watermarking. He joined the Communications and Signal Processing Department, Imperial College London, London, in 2007, where he was a Research Fellow working on 3D object reconstruction. He is currently a Professor with Kingston University, London, working on computer vision and AI for crowd and human behavior analysis, computer games, entertainment, and medical applications. Also, research is conducted on educational games and on HCI for augmented and virtual reality (AR/VR) systems.

Dr. Georgios Th. Papadopoulos is an Assistant Professor in the area of Computer Graphics and Computational Vision at the Department of Informatics and Telematics of the Harokopio University of Athens in Greece. He received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece. He has worked as a Post-doctoral Researcher at the Foundation For Research And Technology Hellas / Institute of Computer Science (FORTH/ICS) and the Centre for Research and Technology Hellas / Information Technologies Institute (CERTH/ITI). He has published over 70 peer-reviewed research articles in international journals and conference proceedings. His research interests include computer vision, artificial intelligence, machine/deep learning, human action recognition, human-computer interaction and explainable artificial intelligence. Dr. Papadopoulos is a member of the IEEE and the Technical Chamber of Greece.

Prof. Panagiotis Sarigiannidis is the Director of ITHACA Lab, Co-Founder of MetaMind Innovations P.C. and Full Professor at the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece. He received his B.Sc. and Ph.D. in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. His research interests include telecommunication networks, Internet of Things and cybersecurity. He has published over 270 papers in international journals, conferences and book chapters, while he has also received five best paper awards. He is involved in several national and international projects. He served as the project coordinator of three H2020 projects, namely SPEAR, EVIDENT and TERMINET. Moreover, he has coordinated national and Erasmus+ KA2 projects, while he served as a principal investigator in SDN-microSENSE and three Erasmus+ KA2: ARRANGE-ICT, JAUNTY and STRONG. Finally, he participates in several editorial boards of various journals.

George K. Karagiannidis (IEEE Fellow) received the Ph.D. degree in Telecommunications Engineering from Electrical Engineering Department, University of Patras, Greece, in 1998. He is currently a Professor with the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki, Greece, and the Head of Wireless Communications and Information Processing (WCIP) Group. His research interests are in the areas of wireless communications systems and networks, signal processing, optical wireless communications, wireless power transfer, and signal processing for biomedical engineering.

Dr. Karagiannidis recently received three prestigious awards: The 2021 IEEE ComSoc RCC Technical Recognition Award, the 2018 IEEE ComSoc SPCE Technical Recognition Award, and the 2022 Humboldt Research Award from Alexander von Humboldt Foundation. He is one of the Highly Cited Authors across all areas of Electrical Engineering, recognized from Clarivate Analytics as the Web-of-Science Highly-Cited Researcher in the ten consecutive years 2015–2024. Currently, he is the Editor-in Chief of IEEE Transactions on Communications and in the past was the Editor-in Chief of IEEE Communications Letters.