A Federated Calibration Framework for ML-based Intrusion Detection Systems

Jacopo Talpini, Nicolò Civiero, Fabio Sartori, and Marco Savi (✉)

Department of Informatics, Systems and Communication
University of Milano-Bicocca
Milano, Italy
{name.surname}@unimib.it

Abstract. Intrusion Detection Systems (IDSs) are essential for protecting connected IoT devices from attacks that threaten network and device integrity, confidentiality, and availability. Machine learning (ML) models are increasingly adopted for IDSs due to their ability to process large volumes of traffic and detect complex threats. However, traditional ML approaches typically rely on centralized data collection, raising privacy and scalability concerns. Federated learning (FL) addresses these issues by enabling collaborative model training across distributed clients without requiring raw data exchange. While much of the existing research on FL-based IDSs focuses on improving accuracy and detection rates, the equally critical aspect of model uncertainty has received little attention. Reliable uncertainty estimation, often achieved through proper model calibration, is crucial for trustworthy decision-making in safety-critical applications such as intrusion detection. This paper proposes a novel federated calibration framework that enables clients to collaboratively train a calibration module while preserving data privacy. Calibration is performed without sharing any local calibration datasets, using a lightweight and efficient method compatible with the classical FL workflow. Experimental results demonstrate that our approach improves model's confidence estimation, while maintaining high predictive performance, making it particularly well-suited for trustworthy IDS deployments in IoT networks.

Keywords: Network Intrusion Detection · Federated Learning · Machine Learning · Model Calibration.

1 Introduction

Network intrusion detection stands as a key pillar of defense against threats to modern communication systems, whose frequency and complexity are steadily increasing [11]. Such intrusions can lead to data breaches, service disruptions, and a significant loss of trust in digital infrastructures. Within this context, the *Internet of Things* (IoT) is rapidly expanding, enabling the seamless integration of heterogeneous devices and computational resources over the Internet, especially at the edge [18].

However, the continued expansion of IoT systems, often consisting of a large number of interconnected and possibly resource-constrained devices, further amplifies the risk of cyber-attacks [32]. The diversity and scale of these networks increase their vulnerability, making them attractive targets for malicious actors. As a result, developing

robust detection strategies and resilient countermeasures is increasingly critical. Proactively identifying vulnerabilities and deploying adaptive defense mechanisms are essential steps toward securing IoT networks [1].

Intrusion Detection Systems (IDSs) play a fundamental role in safeguarding networks by monitoring and identifying malicious activities [14]. Traditionally, intrusion detection has relied on *knowledge-based* systems, which use predefined rules and expert-crafted signatures to recognize known threats [13]. While effective in static environments, these approaches struggle with the dynamic and increasingly complex nature of modern networks, often leading to higher false positive and false negative rates [33, 39]. To overcome these limitations, *data-driven* approaches, particularly those based on machine learning (ML), have emerged as powerful alternatives [29]. ML-based IDSs automatically extract patterns of both normal and malicious behavior from data, often achieving higher detection accuracy across various types of attacks [29]. This paradigm shift is especially relevant in the context of IoT environments, where the high volume, variability, and heterogeneity of data introduce additional challenges [2]: designing effective IDSs in such settings requires robust algorithms capable of generalizing across diverse devices, communication protocols, and usage patterns.

A significant limitation of many current ML-based intrusion detection approaches is their reliance on *centralized* training frameworks, where data collection and model training are conducted on a single central node, typically a server. This centralized paradigm introduces several challenges. Firstly, it demands substantial computational resources concentrated at the central node, which can lead to scalability issues as the volume of data grows. Secondly, the aggregation of vast amounts of training data in one location often results in prolonged training times and increased communication overhead, and also arises concerns about the privacy of users' data [22]. These challenges motivate the exploration of alternative *decentralized* or *distributed* learning paradigms to enhance scalability, efficiency, and privacy.

To tackle those challenges, *federated learning* (FL) was first introduced in [21] and has recently gained traction as a promising training paradigm for ML-based IDSs [7, 36]. FL enables model training directly on decentralized devices, thereby adhering to the principles of *focused collection* and *data minimization*, making it well-suited to reduce many of the systemic privacy risks and operational costs associated with traditional centralized machine learning. In particular, it enables efficient communication and supports low-latency data processing by keeping data localized at the source [16].

Additionally, it is crucial to emphasize that network intrusion detection is a safety-critical application, where the consequences of incorrect predictions, such as falsely classifying an attack or misidentifying the attack type, can be severe and costly. Network administrators require models that are not only *accurate* but also *trustworthy*, primarily in the sense of being able to indicate when their predictions are likely to be incorrect. In particular, the ability to quantify and communicate uncertainty is essential for informed decision-making and timely responses in high-risk security environments. Several studies in the literature have highlighted the pressing need for *trustworthy* (or *reliable*) models specifically tailored to this domain [37], underscoring the importance of integrating reliability measures alongside predictive performance.

A key factor in enhancing the trustworthiness of a machine learning model is its *calibration*, i.e., the degree to which the predicted probabilities reflect the true likelihood of correctness. A model is considered well-calibrated when the confidence it assigns to its predictions corresponds closely to the actual frequency of correct outcomes [12]. For instance, imagine an Internet Service Provider offering an IDS based on a ML model that flags network traffic as either malicious or benign. If the model is calibrated, then among all predictions made with a given confidence level, say, 90%, we expect roughly 90% of samples to be correctly classified. This alignment between confidence and accuracy allows users to meaningfully interpret the model's confidence, and as a result it can be trusted in its classification activity.

Unfortunately, calibrating a model is not always straightforward. Typically, calibration is performed *after* training using a dedicated *calibration dataset* [6]. While this approach works well in centralized settings, where a large amount of data is available both for training and calibration, it may become impractical in privacy-preserving contexts, such as those targeted by federated learning, where data must remain localized and cannot be shared. Conversely, performing calibration solely at a local level on a globally trained model through FL may yield sub-optimal performance.

To address this challenge, we developed a novel *federated calibration* module based on the widely-used calibration technique known as *Platt scaling* [6]. Our approach fully embraces the principles of federated learning and can be applied to any pre-trained model. Due to its design, it is particularly well-suited for federated environments, where calibration itself, like the training process, leverages data locally available at each client. Notably, this module can be integrated in a straightforward way into federated workflows without requiring any modifications to the standard *FedAvg* aggregation algorithm [21], or necessitating changes or retraining of the underlying base model.

We assessed the effectiveness of our proposed approach using the *ToN-IoT* dataset [31], a widely recognized benchmark for analyzing attacks on IoT infrastructures [3]. Our evaluation particularly focused on how data heterogeneity across clients impacts the calibration performance of the method. For a comprehensive comparison, we included several baselines: a classifier trained within a federated learning framework *without calibration*, which exhibited poor calibration properties; a *centralized calibration* approach, where the calibration module is trained using aggregated data collected from clients at a central location; and a personalized *local calibration* approach, where each client independently trains its own calibrator.

Our results demonstrate that model calibration effectively enhances reliability. In particular, the proposed federated calibration approach achieves *(i)* superior calibration performance compared to local calibration, and *(ii)* only marginally lower performance than that of centralized calibration, while preserving data privacy, unlike the centralized setting. Importantly, this improvement is obtained without significantly impacting the model's classification performance in terms of F1-Score and accuracy.

This paper is an extended version of [35], including additional details and some novel results. The main contributions of the work are twofold, and are summarized in the following:

- We propose a federated calibration module compatible with any pre-trained classifier, specifically designed for federated learning environments and ideal for safety-critical applications.
- We evaluate the proposed approach in the context of IoT network intrusion detection, considering varying levels of data heterogeneity across clients, and compare its performance against several relevant baseline methods.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed federated calibration methodology, while Section 4 outlines the experimental setup and describes the dataset used. Section 5 reports and discusses the numerical results. Finally, Section 6 concludes the paper by summarizing the key findings and insights.

2 Related Work

Intrusion Detection Systems have traditionally served as a secondary layer of network defense, aimed at monitoring traffic and detecting malicious activities that may bypass primary security mechanisms such as firewalls [23]. These systems are broadly classified into two main categories: *signature-based* and *anomaly-based* IDSs [38].

Signature-based IDSs, also referred to as *misuse* detection systems, rely on predefined patterns or signatures of known attacks to detect intrusions. They are effective at identifying previously encountered threats but struggle with zero-day attacks or novel intrusions. On the other hand, *anomaly-based* IDSs construct a model of normal behavior and flag deviations from this learned baseline as potential threats. While they are better suited to detect previously unseen attacks, they often suffer from a high false positive rate, which can lead to unnecessary alerts and increased operational overhead [2].

In this work, we focus on signature-based intrusion detection, specifically within the context of IoT environments. The following subsection reviews key contributions in the area of data-driven systems based on machine learning applied to intrusion detection. After that, we also highlight recent advancements aimed at improving model calibration in federated learning frameworks.

2.1 ML-based Intrusion Detection Systems

In recent years, data-driven approaches have gained significant traction in the development of IDSs [33, 19, 38]. Various machine learning techniques, such as random forests, support vector machines, neural networks, and clustering algorithms, have been explored to automatically learn patterns indicative of malicious activity. Among these, machine learning and deep learning methods have emerged as particularly effective, thanks to their ability to model complex and nonlinear relationships in high-dimensional data. This has proven especially valuable in IoT environments, where traffic patterns are often heterogeneous and traditional rule-based systems fall short [33, 8, 30]. However, most intrusion detection approaches are built upon centralized architectures, where locally collected data is transmitted to cloud datacenters or centralized servers. These systems capitalize on the considerable computational resources available at the central

nodes to train detection models [7]. While effective in terms of performance, such centralized frameworks raise concerns related to scalability, latency, and data privacy. In response to these challenges, the federated learning paradigm has emerged as a valid alternative, enabling collaborative model training without requiring raw data to leave local devices. Although still an emerging area, a growing body of research has begun to explore the feasibility and benefits of FL for intrusion detection, also in relationship to IoT scenarios [27, 7, 4, 28, 36].

A representative example of applying federated learning to intrusion detection is presented in [28], where the authors design a FL-based framework for malware detection by combining both supervised and unsupervised learning techniques. In particular, they utilize a multi-layer perceptron to perform binary classification and an autoencoder to identify anomalies. Their experimental setup assumes a balanced data distribution across clients, meaning that each participant holds similar class proportions: however, this is a condition that rarely holds in real-world IoT deployments. In contrast, the study conducted in [7] takes a more realistic approach by evaluating FL in the context of IoT using a heterogeneous dataset. The authors emphasize the detrimental effects of statistical heterogeneity, a common challenge in federated environments, where each device may observe different types and frequencies of network attacks. This variability in local data distributions undermines the effectiveness of traditional FL algorithms, which typically assume data homogeneity, and necessitates the development of specialized techniques to ensure reliable model performance across diverse clients.

Another contribution falls in the domain of trustworthy intrusion detection [37]. In that work we demonstrated how uncertainty quantification can significantly improve the reliability of IDSs. Such a study, however, was conducted within the conventional centralized learning framework. This work aligns with the vision outlined in [37], but advances it by proposing a federated calibration approach that enables the deployment of well-calibrated models across distributed clients, which is a primary requirement for an uncertainty-aware model.

2.2 Model Calibration in a Federated Scenario

Calibration has been widely investigated in the literature, particularly within the traditional centralized setting, where machine learning models, especially deep neural networks, are often found to be poorly calibrated (see [12] for a comprehensive overview). Techniques to address miscalibration can generally be categorized into two groups: methods that intervene during the training phase, such as modifying the model architecture or employing alternative loss functions (e.g. focal loss), and post-hoc approaches that operate after training, typically requiring a separate calibration dataset disjoint from the training data. Among the most prominent post-hoc techniques are temperature scaling, Platt scaling, and isotonic regression, which adjust the model's confidence scores without altering the underlying classifier. These methods are attractive due to their simplicity and compatibility with pre-trained models [12].

Despite its critical role in reliable decision-making, model calibration in FL setting has so far received relatively limited attention. Among the few notable efforts, recent research [25] highlighted that conventional FL algorithms often yield poorly calibrated models after the standard weight aggregation step. This miscalibration raises concerns,

particularly for deploying FL models in safety-critical domains as those of IoT network intrusion detection. To address this issue, the authors proposed a post-hoc calibration strategy based on a multi-layer perceptron trained on client data. While effective in improving calibration, the method relies on a large number of learnable parameters, making it prone to overfitting in the case of calibration sets of modest size and introduces considerable communication overhead between clients and the central server, potentially limiting its practicality in real-world federated settings.

In a related effort, [26] addressed the challenge of model calibration in federated learning by introducing a method that uses prototype-based representations of each client's data clusters to enable effective calibration. However, their approach is tailored to the cross-silo FL setting, where clients can engage in peer-to-peer communication. By contrast, this work targets a cross-device FL scenario, in which clients interact solely with a central server, making direct client-to-client communication infeasible.

Another recent study [10] introduced a modification to the local training loss by adding a calibration loss term to encourage improved model calibration. Although this approach is effective for training neural networks from scratch, it is not applicable in scenarios involving pre-trained classifiers, as considered in this paper, since it relies on direct access to the training procedure.

All the previous studies address model calibration in a general context, without targeting any specific application domain. To the best of our knowledge, the challenge of achieving reliable model calibration while performing federated learning specifically for IoT network intrusion detection has not yet been investigated. This paper aims to bridge that gap.

3 Our Approach: Federated Calibration

As previously discussed, proper calibration is essential for achieving reliable ML models. While model trustworthiness has been mainly studied in centralized settings, we introduce a *federated calibration module* (or *federated calibrator*) designed to work with any pre-trained model. This module can be integrated into federated learning workflows without requiring changes to the standard FedAvg [21] aggregation protocol. Specifically, our approach is novel as the model is not only trained in a federated manner, but also calibrated collaboratively, leveraging local calibration data from individual clients to improve its overall reliability.

The primary objective of the calibrator is to transform the raw predictive scores produced by any classifier into well-calibrated probabilities that accurately reflect the true likelihoods of each class. Specifically, given a classifier that outputs predictive scores across C classes $[f_0, .., f_{C-1}]$ (i.e., benign or specific attacks) for a given input $\boldsymbol{x}$ (i.e., network traffic sample), establish a meaningful mapping transforming these scores into well-calibrated probabilities. To accomplish this, we adopt *Platt scaling*, also referred to as *sigmoid* (or logistic) *calibration*. Platt scaling is an appealing choice due to its simplicity, parametric form, and proven effectiveness in many calibration tasks. Moreover, its parametric nature aligns well with the constraints and communication efficiency requirements of federated learning workflows, making it a natural fit for decentralized calibration across multiple clients.

More specifically, suppose a *binary classification* problem (for simplicity and without any loss of generality) and assume that our base classifier (e.g. a neural network or a decision tree) predicts some score for the 0-*th* class, denoted as $f_0(\boldsymbol{x})$, for a given input $\boldsymbol{x}$. Platt scaling maps this score to a probability through a *sigmoid function* characterized by two learnable parameters, denoted here as a and b, as follows:

$$p(y=0|f_0(\boldsymbol{x})) = \text{sigmoid}(a \cdot f_0(\boldsymbol{x}) + b) \tag{1}$$

Those parameters are estimated through maximum likelihood on a validation set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ of input-output pairs ($\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, respectively) that, in this context, we call *calibration dataset*. In the case of a classification task (i.e., the task considered in this paper), the loss function reduces to the usual cross-entropy loss between the predicted probabilities for a given input, and the corresponding class label.

The formulation in Equation (1) can be straightforwardly extended to multi-class classification by applying the calibration function independently to each class's output scores and then normalizing the results to produce a valid probability distribution. One of the key advantages of Platt scaling is its simplicity: it involves only a small number of parameters that grows linearly with the number of classes, making it particularly well-suited for calibration tasks even when limited data is available. Furthermore, its versatility allows it to be applied to any pre-trained model without requiring changes to the original training process, making it an ideal choice for federated learning scenarios.

Our straightforward yet effective proposal is to *train the calibrator within the standard federated learning approach*, for example, by employing the FedAvg algorithm [21] on a centralized server. Specifically the server, acting as the federated model aggregator, distributes a predefined calibrator model to the clients (or a randomly selected subset of them). This model is parameterized by a and b, which characterize the sigmoid function used in Platt scaling. Each client then performs local training of the calibrator using its own calibration data, optimizing a given loss function (typically cross-entropy) and computes an updated set of model parameters. These local updates are related to $\boldsymbol{w} \equiv [a, b]$, following the formulation in Equation (1).

At the end of each communication round, the central server collects the locally updated Platt scaling parameters a and b from the participating clients. These updates are then aggregated to form a new global version of the calibrator, which is subsequently shared in the next round with clients. This iterative procedure allows a potentially large number of clients to collaboratively refine the calibration function without ever transmitting their local calibration datasets, since only the learned sigmoid parameters need to be communicated.

As previously discussed, the aggregation mechanism employed in this work is FedAvg, which computes the global parameter update $\boldsymbol{w}_{\text{global}}$ as a weighted average of the local parameter vectors $\boldsymbol{w}_k = [a_k, b_k]$ provided by each client:

$$\boldsymbol{w}_{\text{global}} = \frac{1}{\sum_{k=1}^{K} n_k} \cdot \sum_{k=1}^{K} n_k \boldsymbol{w}_k. \tag{2}$$

This strategy maintains the decentralized nature of federated learning while ensuring that the calibration process benefits from diverse, distributed data sources, while keeping data decentralized.

Algorithm 1 Federated Calibrator (from [35])

Require: Pre-trained classifier, $\tilde{\boldsymbol{y}} = f(\boldsymbol{x})$, and the calibration module $\hat{\boldsymbol{y}} = \text{calibrator}(\tilde{\boldsymbol{y}}, \boldsymbol{w})$
Require: For each client k, a fresh calibration dataset $\mathcal{D}_k = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_k}$ and the corresponding predictions provided by the base classifier $\{\tilde{\boldsymbol{y}}_i\}_{i=1}^{N_k}$
Require: M federated training rounds, learning rate η
  Server randomly initializes global model weights $\boldsymbol{w}^1$
  for $m = 1, .., M$ **do**
    for $k = 1, .., K$ **do**
      $\{\hat{\boldsymbol{y}}_i = \text{calibrator}(f(\boldsymbol{x}_i), \boldsymbol{w}^m)\}_{i=1}^{N_k}$ ▷ calibrator predictions
      $\mathcal{L}_k(\boldsymbol{w}) = \sum_{i=1}^{N_k} \text{cross-entropy}[(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)]$ ▷ local loss estimation
      $\boldsymbol{w}_k^m \leftarrow \boldsymbol{w}_k^m - \eta \nabla \mathcal{L}_k(\boldsymbol{w})$ ▷ weights update
    end for
    Server aggregates clients' updates:
    $\boldsymbol{w}_{\text{global}}^{m+1} \leftarrow \text{FedAvg}(\{\boldsymbol{w}_k^m, N_k\}_{k=1}^{K})$ ▷ Eq. (2)
  end for
  return $\boldsymbol{w}_{\text{global}}^{M}$ ▷ calibrator parameters global estimate

In summary, the entire calibration process is orchestrated by the central server, which takes a pre-trained *base model* as input and produces its calibrated counterpart as output. This is achieved by executing the federated calibration module, which operates according to the procedure outlined in Algorithm 1.

It is worth noting once again that the base model to be calibrated can originate from any training paradigm, whether centralized, decentralized, or federated. However, due to the design and operational characteristics of our calibration framework, the proposed federated calibration procedure is particularly well-suited for models that have been trained using federated learning. This is because it naturally aligns with the same communication and aggregation workflow, ensuring seamless integration into existing FL pipelines. In practice, the model can first be trained across distributed clients using a standard federated learning algorithm, and subsequently passed as input to our federated calibration module for post-hoc calibration.

4 Dataset and Experimental Setup

4.1 Dataset Description

To explore the feasibility and effectiveness of our approach, the selection of a realistic and representative dataset plays a crucial role. As highlighted in [7], there exists an extensive review of publicly available datasets tailored to intrusion detection in the IoT domain. Following the methodology adopted in that work and considering the desirable properties for our use case, we rely on the *NF-ToN-IoT dataset* [31] *version 2*, which builds upon the original *ToN_IoT* dataset [3]. This dataset has been specifically designed to capture network traffic scenarios relevant to IoT environments, making it a suitable benchmark for our federated learning experiments.

Each entry in the dataset corresponds to a network flow, described by 43 numerical and categorical features, and is annotated with one of 10 possible class labels, compris-

ing benign traffic and nine distinct types of attacks. The dataset is highly imbalanced, with approximately 17 million total samples, of which 64% are labeled as attacks and the remaining 36% as benign. To distribute the data among clients, we adopted a widely used partitioning strategy based on the symmetric *Dirichlet distribution*, controlled by a concentration parameter α [15]. This parameter serves as a proxy for the level of *statistical heterogeneity* in the data: lower values of α (e.g. $\alpha < 1$) result in more uneven class distributions across clients, thereby simulating non-IID conditions commonly encountered in real-world settings. In our experiments, we simulated a federation of 10 clients and explored different degrees of heterogeneity by varying α in the range $[0.2, 0.9]$.

It is worth further emphasizing that statistical heterogeneity is a typical characteristic of malicious traffic in IoT networks, as noted in [7]. Motivated by this observation, we focused our analysis primarily on heterogeneous settings ($\alpha < 1$) to better reflect realistic deployment conditions and evaluate the robustness of our approach under challenging data distributions.

Moreover, to take advantage of the simplicity and data efficiency of the calibration module, a *sampling strategy* to balance the class distribution during calibration has been applied. Specifically, for each client, we randomly reduced the number of samples from majority classes while retaining all available instances from the minority class. This ensures that all classes are equally represented during the calibration phase, helping to mitigate bias and improve calibration performance across under-represented categories. More specifically, we considered two scenarios, where the calibration set contains 50 samples per class or 1000 samples per class.

4.2 Adopted Classifier and Calibrator Implementation

For our base model, we selected an extreme gradient-boosted decision tree classifier, implemented using the *XGBoost* library [9]. This choice is motivated by the well-established effectiveness of XGBoost on tabular datasets, which are common in network traffic analysis. In fact, gradient-boosted trees often outperform more complex models, such as deep neural networks, when working with structured data of this nature [34, 20]. This makes XGBoost a particularly good choice for the network traffic classification task addressed in this work.

In the proposed federated setting, each client independently trains its own XGBoost model on local data. These locally trained models are then transmitted to a central server, which aggregates them to form a global model in accordance with the FL paradigm. Specifically, for boosted decision trees, the aggregation process proceeds as follows: at each communication round, clients send their trained boosted trees to the server, which interprets them as bootstrapped instances of a classifier. The server then combines these trees into an ensemble, effectively constructing the final global model. More concretely, if we have K clients and M rounds, the final classifier will consist of $K \cdot M$ trees. The model aggregation procedure is performed exploiting the Flower library [5], and the FL training of the tree-based model required 50 training rounds and 10 local epochs per round.

For the calibration module, Platt scaling was implemented using PyTorch [24] as a fully connected neural network with no hidden layers, a diagonal weight matrix, and

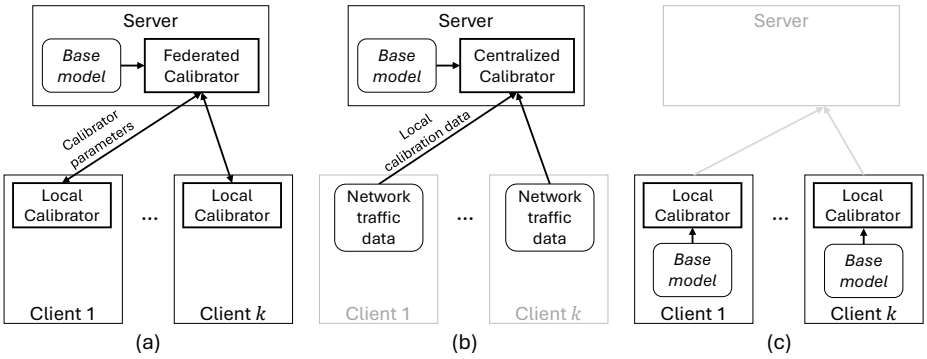

Fig. 1: Pictorial view of our federated calibration approach (a), centralized calibration (b), and local calibration (c) (from [35]).

a sigmoid activation function, corresponding to the Platt scaling formulation in Equation (1). This straightforward structure enabled seamless integration of the calibration module into the Flower federated learning framework. Federated training of the calibrator was carried out over 20 communication rounds, with each client performing 10 local training epochs per round.

4.3 Baseline Strategies

To assess the effectiveness of the proposed federated calibration approach, we compared the following strategies:

- **Base model** (no calibration). This is the classifier obtained by federated learning without any calibration module, as described in Section 4.2.
- **Centralized calibration**. This setup represents the most favorable scenario for calibration performance, as the central server has direct access to a dedicated calibration dataset, allowing calibration to be carried out in a traditional centralized manner. However, this approach compromises data privacy, since it requires clients to share a portion of their local data with the central server.
- **Local calibration**. Each client trains its calibrator using exclusively and independently its local data. The result of this operation is that the calibration is personalized for each client.
- **Proposed approach** (federated calibration). This corresponds to our novel federated calibration method introduced in Section 3.

A high-level comparison of the different calibration approaches is shown in Figure 1. All the model parameters, such as the number of local training epochs and the number of federation training rounds, are kept fixed to the values specified in Section 4.2 across the different strategies and across each value of α, to ensure a fair comparison of the models' performance.

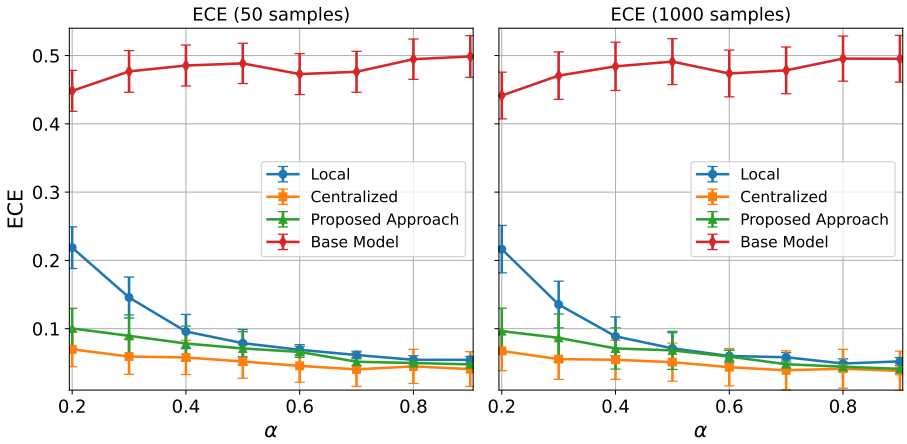

Fig. 2: Expected Calibration Error for the considered models as a function of the clients' data statistical heterogeneity and per-class number of samples. The curves represent the mean ECE along with the 95% confidence intervals, calculated over 20 experiments with different random seeds (partially from [35]).

5 Performance Evaluation

5.1 Evaluated Metrics

The primary objective of this paper is to enhance the calibration of a given classifier. A widely used metric to evaluate model calibration is the *Expected Calibration Error* (ECE) [12]. ECE measures the difference between predicted confidence and actual accuracy by partitioning the predicted probabilities into discrete bins. For each bin, it compares the average predicted probability with the empirical accuracy, the fraction of correctly classified samples whose predicted confidence falls within that bin, quantifying the overall discrepancy as an indicator of calibration quality. More precisely, it is defined as:

$$\mathrm{ECE} = \frac{|\mathcal{B}_m|}{N_s} \sum_{n=1}^{N_b} |\mathrm{acc}(\mathcal{B}_n) - \mathrm{conf}(\mathcal{B}_n)| \tag{3}$$

where N_b is the number of bins (typically 10), N_s is the total number of samples, *acc* and *conf* represent the accuracy and the confidence for the samples belonging to the n-th bin, denoted as $\mathcal{B}_n$. ECE values range in the range $[0,1]$, where lower values indicate better model calibration.

In addition, we also evaluated the resulting models with standard predictive performance metrics such as *accuracy* and *F1-Scores*, including both the Macro F1-Score (*F1-Macro*) and the Weighted F1-Score (*F1-Weighted*), which accounts for the class imbalance by weighting each class's contribution proportionally to its sample size. A desirable outcome of calibration is minimizing the ECE of a classifier without adversely impacting classification performance.

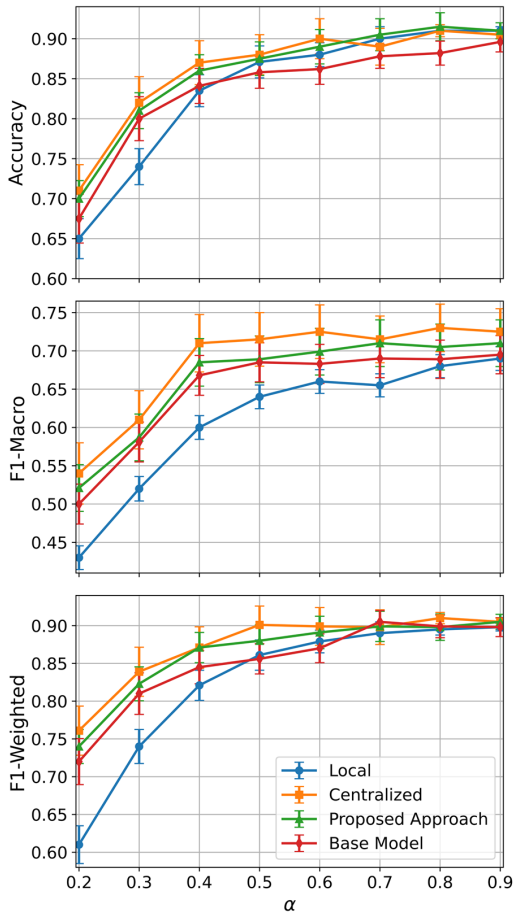

Fig. 3: Accuracy and F1-Scores for the considered models as a function of the clients' data statistical heterogeneity. The curves represent the mean values of each metric along with the 95% confidence interval, computed over 20 different experiments with varying random seeds (from [35]).

Finally, we evaluated the *communication costs* of the different calibration strategies, expressed as the amount of data that needs to be delivered from the clients to the server and vice versa to make the strategy working (see Figure 1).

5.2 Calibration and Classification Performance

Figure 2 shows the ECE as a function of the heterogeneity parameter α for the models obtained as an output by the considered strategies. The analysis is performed taking into consideration two different sizes of the calibration dataset: 50 or 1000 samples per class. Overall, the results show that the base model is significantly miscalibrated across

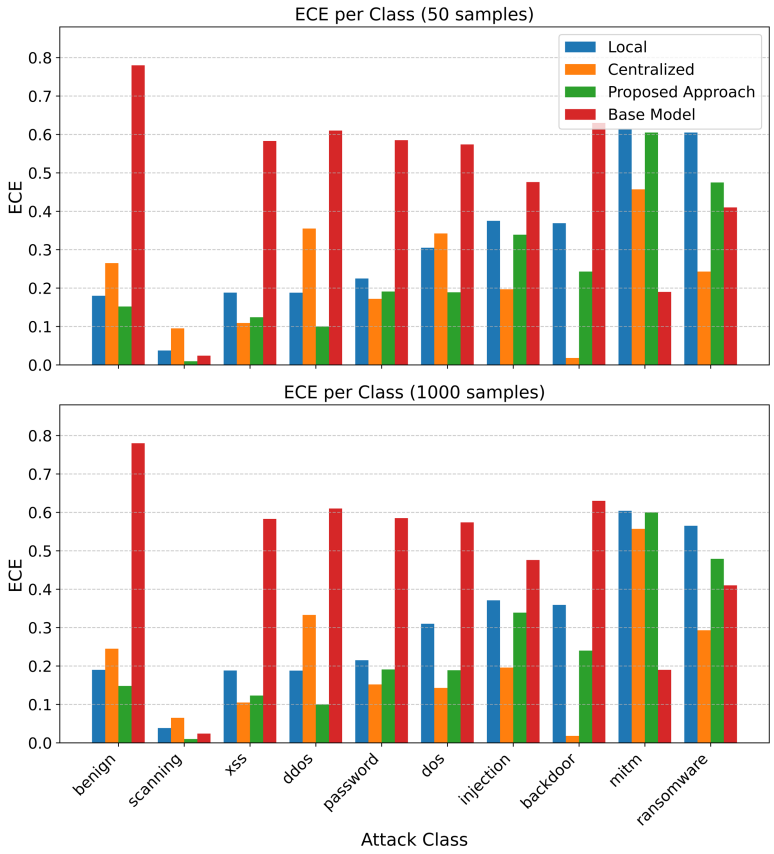

Fig. 4: Per-class ECE for the evaluated models with fixed data heterogeneity ($\alpha = 0.3$) and for different calibration set sizes (50 and 1000 samples per class). The classes are ordered in decreasing order based on their sample size. The ECE represents a mean value, computed over 20 different experiments with varying random seeds (partially from [35]).

all values of α, highlighting potential reliability issues in standard FL-based models. In contrast, the proposed approach achieves calibration performance comparable, and only slightly worse, to that of centralized calibration, without requiring access to a centralized calibration dataset, thereby ensuring both privacy preservation and computational efficiency by keeping data local.

Additionally, the proposed approach achieves better calibration compared to personalized, local calibration methods. The difference, as expected, is higher in more heterogeneous settings (i.e., with low α), where local data are not representative of the overall population, as typically happens in an IoT network intrusion detection scenario. For example, certain regions of an IoT network might be more susceptible to specific cyberattacks than others, resulting in imbalanced data distributions across clients. En-

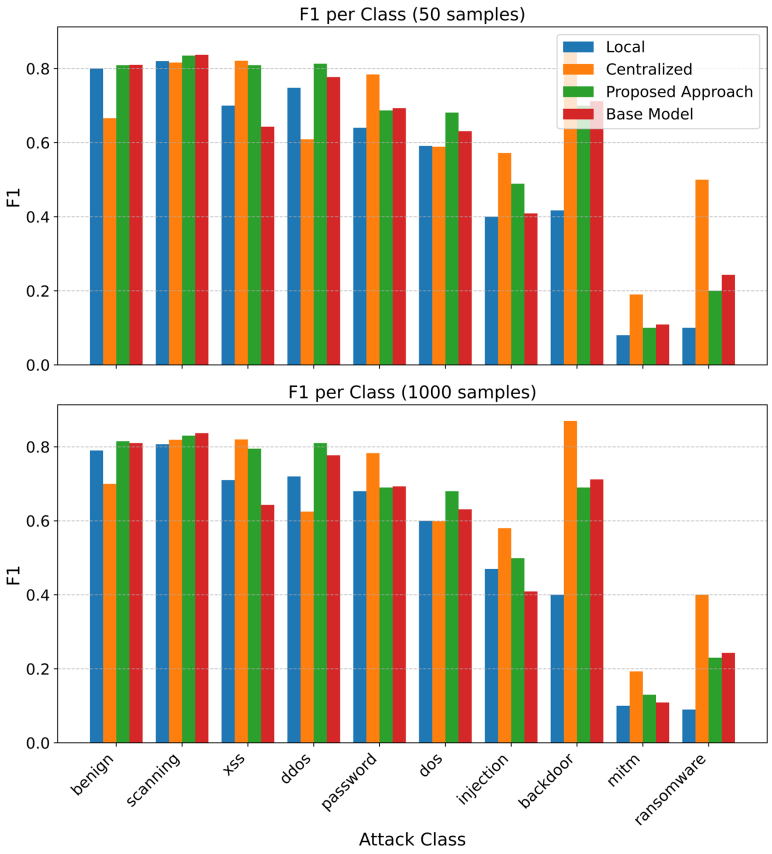

Fig. 5: Per-class F1-Score for the evaluated models with fixed data heterogeneity ($\alpha = 0.3$) for different calibration set sizes (50 and 1000 samples per class). The classes are ordered in decreasing order based on their sample size. The F1-Score represents a mean value, computed over 20 different experiments with varying random seeds.

suring robust model performance under such conditions is crucial for the practical deployment and effectiveness of large-scale Intrusion Detection Systems. Moreover, the calibration module demonstrates robustness to the size of the calibration set, largely due to the simplicity of Platt scaling. Increasing the calibration set size by a factor of 20 (i.e., by considering 1000 samples per class instead of only 50) yields to comparable calibration performance, indicating that calibration by means of Platt scaling is effective even in data-scarce regimes, and indiscriminately adding calibration data does not positively impact on calibration performance.

It is also important to highlight that the calibration module does not degrade the classification performance. As illustrated in Figure 3, our federated calibration approach maintains accuracy, F1-Macro, and F1-Weighted scores comparable to both the centralized calibration scenario and the original base model. This demonstrates that improving

model confidence through calibration does not come at the cost of worsening the predictive performance. In contrast, local calibration leads to noticeably inferior results, particularly in highly heterogeneous settings.

Finally, to further investigate the relationship between ECE and class imbalance, we report in Figure 4 the per-class ECE, with classes sorted in decreasing order of sample size, for a fixed heterogeneity parameter $\alpha = 0.3$ and across different calibration set sizes (i.e, 50 and 1000 samples per class). The results highlight that minority classes (e.g. ransomware, mitm, etc.) exhibit significantly higher ECE values, indicating a stronger calibration challenge in these cases for all the calibration approaches, showing how Platt scaling struggles in performing its task. Moreover, results are comparable for the two considered calibration dataset sizes, with noticeable differences only for the minority classes, where not surprisingly ECE is lower when 1000 samples are considered rather than 50.

To further investigate performance degradation related to minority classes, we present in Figure 5 the per-class F1-Score, again with classes ordered by decreasing sample size. This analysis complements the ECE trends by showing that under-represented classes tend to exhibit the lowest predictive performance, and that our federated approach tends to perform worse on these classes compared to a centralized approach.

5.3 Communication Costs

One of the main benefits of a federated calibration approach is its computational efficiency, arising from the inherent parallelism of distributed data processing, as well as its communication efficiency. In this section we focus on the latter, comparing communication costs for the different calibration strategies, and expressing them in bytes:

- **Centralized calibration**. The communication costs scale linearly with the size of the dataset and the number of considered features. In our experimental setup, the communication cost is approximately ~ 84 KB, representing the size of the calibration dataset to be collected by the server from the clients.
- **Local calibration**. The local calibrator does not incur any communication costs as no data needs to be shared with the server. However its performance is low, as seen in the previous sections.
- **Proposed approach** (federated calibration). The communication costs scale linearly with the size of the model and the number of communication rounds. Given the simplicity of Platt scaling, the communication costs are significantly lower than that of the centralized calibration, i.e., approximately ~ 3.9 KB considering our experimental setup. These costs are related to the model's parameters (i.e., a and b) to be exchanged between clients and server in the different communication rounds.

6 Conclusion

In this paper, we introduced a novel federated calibration module that leverages a lightweight calibrator trained using the standard federated learning pipeline. The proposed approach achieves strong calibration performance even in challenging scenarios

with high data heterogeneity (i.e., low α values), and maintains results comparable to centralized, server-side calibration as α increases (i.e., $\alpha > 0.4$). Crucially, it offers significant privacy benefits, since no calibration data is shared with the central server. Additionally, our federated calibration consistently outperforms a local calibration strategy, in which each client independently calibrates the base model (e.g. one obtained through FL). This local approach suffers from limited data representativeness on individual nodes, which can lead to poorly calibrated and unreliable models. In contrast, our method ensures consistency and reliability without compromising data privacy. In addition, our federated calibration does not compromise classification performance: accuracy and F1-Score remain comparable with the uncalibrated base model and are similar to those achieved by a centrally calibrated model.

These properties make our approach especially suitable for deploying reliable ML-based Intrusion Detection Systems in IoT networks, where data is often imbalanced and preserving privacy and computational efficiency is critical. However, the proposed approach still encounters difficulties when dealing with under-represented classes, indicating a key area for further enhancement. As part of future work, we plan to explore alternative calibration techniques within the federated learning framework, such as isotonic regression and methods inspired by the conformal prediction paradigm [17].

Acknowledgment

The research leading to these results has been partially funded by the Italian Ministry of University and Research (MUR) under the PRIN 2022 PNRR framework (EU Contribution – NextGenerationEU – M. 4,C. 2, I. 1.1), SHIELDED project, ID P2022ZWS82.

References

1. Abdulkareem, S.A., Heng Foh, C., Shojafar, M., Carrez, F., Moessner, K.: Network Intrusion Detection: An IoT and Non IoT-Related Survey. IEEE Access **12**, 147167–147191 (2024)
2. Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I., Guizani, M.: A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. IEEE Communications Surveys Tutorials **22**(3), 1646–1685 (2020)
3. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: TON-IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. IEEE Access **8**, 165130–165150 (2020)
4. Aouedi, O., Piamrat, K., Muller, G., Singh, K.: Intrusion Detection for Softwarized Networks with Semi-supervised Federated Learning. In: IEEE International Conference on Communications (ICC) (2022)
5. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., de Gusmão, P.P.B., et al.: Flower: A Friendly Federated Learning Research Framework. arXiv preprint arXiv:2007.14390 (2020)
6. Böken, B.: On the Appropriateness of Platt Scaling in Classifier Calibration. Information Systems **95**, 101641 (2021)
7. Campos, E.M., Saura, P.F., González-Vidal, A., Hernández-Ramos, J.L., et al.: Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges. Computer Networks p. 108661 (2021)

8. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P.: Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Communications Surveys Tutorials **21**(3), 2671–2701 (2019)
9. Chen, T., Guestrin, C.: Xgboost: A Scalable Tree Boosting System. In: ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD) (2016)
10. Chu, Y.W., Han, D.J., Hosseinalipour, S., Brinton, C.: Unlocking the Potential of Model Calibration in Federated Learning. arXiv preprint arXiv:2409.04901 (2024)
11. European Union Agency for Cybersecurity: ENISA Threat Landscape 2023. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023 (2023), [Accessed: 09-December-2024]
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On Calibration of Modern Neural Networks. In: International Conference on Machine Learning (ICML) (2017)
13. Hassija, V., Chamola, V., Saxena, V., Jain, D., et al.: A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. IEEE Access **7**, 82721–82743 (2019)
14. Heidari, A., Jabraeil Jamali, M.A.: Internet of Things Intrusion Detection Systems: A Comprehensive Review and Future Directions. Cluster Computing **26**(6), 3753–3780 (2023)
15. Hsu, T.M.H., Qi, H., Brown, M.: Measuring the Effects of Non-identical Data Distribution for Federated Visual Classification. arXiv preprint arXiv:1909.06335 (2019)
16. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., et al.: Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)
17. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., Das, S.K.: Edge-computing-driven Internet of Things: A Survey. ACM Computer Surveys **55**(8) (2022)
19. Liu, H., Lang, B.: Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences **9**, 4396 (2019)
20. Markovic, T., Leon, M., Buffoni, D., Punnekkat, S.: Random Forest based on Federated Learning for Intrusion Detection. In: International Conference on Artificial Intelligence Applications and Innovations (AIAI). Springer (2022)
21. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient Learning of Deep Networks from Decentralized Data. In: International Conference on Artificial Intelligence and Statistics (2017)
22. Mothukuri, V., Parizi, R., Pouriyeh, S., Huang, Y., et al.: A Survey on Security and Privacy of Federated Learning. Future Generation Computer Systems **115**, 619–640 (2020)
23. Moustafa, N., Hu, J., Slay, J.: A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey. Journal of Network and Computer Applications **128** (2018)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An Imperative Style, High-performance Deep Learning Library. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
25. Peng, H., Yu, H., Tang, X., Li, X.: FedCal: Achieving Local and Global Calibration in Federated Learning via Aggregated Parameterized Scaler. In: International Conference on Machine Learning (ICML) (2024)
26. Qi, Z., Meng, L., Chen, Z., Hu, H., Lin, H., Meng, X.: Cross-silo Prototypical Calibration for Federated Learning with Non-iid Data. In: ACM International Conference on Multimedia (2023)
27. Rahman, S.A., Tout, H., Talhi, C., Mourad, A.: Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning? IEEE Network **34**(6), 310–317 (2020)
28. Rey, V., Sánchez, P.M.S., Celdrán, A.H., Bovet, G.: Federated Learning for Malware Detection in IoT Devices. Computer Networks **204**, 108693 (2022)

29. Saranya, T., Sridevi, S., Deisy, C., Chung, T.D., Khan, M.: Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. Procedia Computer Science **171**, 1251–1260 (2020)
30. Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., Portmann, M.: Feature Extraction for Machine Learning-based Intrusion Detection in IoT Networks. Digital Communications and Networks **10**(1), 205–216 (2022)
31. Sarhan, M., Layeghy, S., Portmann, M.: Evaluating Standard Feature Sets Towards Increased Generalisability and Explainability of ML-based Network Intrusion Detection. Big Data Research **30**, 100359 (2022)
32. Sasi, T., Lashkari, A.H., Lu, R., Xiong, P., Iqbal, S.: A Comprehensive Survey on IoT Attacks: Taxonomy, Detection Mechanisms and Challenges. Journal of Information and Intelligence **2**(6), 455–513 (2024)
33. Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence **2**(1), 41–50 (2018)
34. Shwartz-Ziv, R., Armon, A.: Tabular Data: Deep Learning is not All You Need. Information Fusion **81**, 84–90 (2022)
35. Talpini, J., Civiero, N., Sartori, F., Savi, M.: A Federated Approach to Enhance Calibration of Distributed ML-Based Intrusion Detection Systems. In: International Conference on Agents and Artificial Intelligence (ICAART) (2025)
36. Talpini, J., Sartori, F., Savi, M.: A Clustering Strategy for Enhanced FL-Based Intrusion Detection in IoT Networks. In: International Conference on Agents and Artificial Intelligence (ICAART) (2023)
37. Talpini, J., Sartori, F., Savi, M.: Enhancing Trustworthiness in ML-based Network Intrusion Detection with Uncertainty Quantification. Journal of Reliable Intelligent Environments **10**(4), 501–520 (2024)
38. Tauscher, Z., Jiang, Y., Zhang, K., Wang, J., Song, H.: Learning to Detect: A Data-driven Approach for Network Intrusion Detection. In: IEEE International Performance Computing and Communications Conference (IPCCC) (2021)
39. Tsimenidids, S., Lagkas, T., Rantos, K.: Deep Learning in IoT Intrusion Detection. Journal of Network and Systems Management **30**(1), 8 (2022)