A Federated Approach to Enhance Calibration of Distributed ML-Based Intrusion Detection Systems

Jacopo Talpini[a], Nicolò Civiero, Fabio Sartori[b] and Marco Savi[c]
Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milano, Italy
{*firstname.lastname*}*@unimib.it*

Keywords: Intrusion Detection, Federated Learning, Machine Learning, Model Calibration.

Abstract: Network intrusion detection systems (IDSs) are a major component for network security, aimed at protecting network-accessible endpoints, such as IoT devices, from malicious activities that compromise confidentiality, integrity, or availability within the network infrastructure. Machine Learning models are becoming a popular choice for developing an IDS, as they can handle large volumes of network traffic and identify increasingly sophisticated patterns. However, traditional ML methods often require a centralized large dataset thus raising privacy and scalability concerns. Federated Learning (FL) offers a promising solution by enabling a collaborative training of an IDS, without sharing raw data among clients. However, existing research on FL-based IDSs primarily focuses on improving accuracy and detection rates, while little or no attention is given to a proper estimation of the model's uncertainty in making predictions. This is however fundamental to increase the model's reliability, especially in safety-critical applications, and can be addressed by an appropriate model's calibration. This paper introduces a federated calibration approach that ensures the efficient distributed training of a calibrator while safeguarding privacy, as no calibration data has to be shared by clients with external entities. Our experimental results confirm that the proposed approach not only preserves model's performance, but also significantly enhances confidence estimation, making it ideal to be adopted by IDSs.

1 INTRODUCTION

Network intrusions represent a significant threat to modern communication systems, with their frequency and complexity steadily increasing (European Union Agency for Cybersecurity, 2023). These incidents compromise data, disrupt services, and erode trust in digital infrastructures. In this landscape, the *Internet of Things* (IoT) represents a growing paradigm that facilitates the connection of diverse devices and computational capabilities through the Internet.

However, the ongoing expansion of IoT systems, which often involve a substantial number of devices, heightens even more the risk of cyber-attacks. As a result, the development of effective detection strategies and resilient countermeasures has become critical: proactively identifying vulnerabilities and implementing adaptive defense mechanisms are essential to safeguarding IoT networks (Hassija et al., 2019).

Intrusion Detection Systems (IDSs) play a primary role to accomplish this task (Khraisat et al., 2019). Traditional intrusion detection methods have mainly relied on *knowledge-based systems* (Hassija et al., 2019). However, as networks become more complex, these methods are increasingly prone to errors (Shone et al., 2018; Tsimenidis et al., 2022). To address these challenges, *data-driven approaches* leveraging *machine learning (ML)* have gained significant attention in recent years (Saranya et al., 2020), with a strong focus on detecting attacks in IoT environments (Al-Garadi et al., 2020), which is difficult given the high data heterogeneity.

However, a limitation of current ML-based approaches is their reliance on centralized training, where data and computational resources are processed and managed by a single central node, such as a server. Such a centralized framework is associated with several challenges, including high computational requirements, extended training times, and heightened concerns over the security and privacy of users' data (Mothukuri et al., 2020).

To address these issues, *federated learning (FL)* was originally proposed in (McMahan et al., 2017) and has recently emerged as an effective model train-

[a] https://orcid.org/0000-0003-1556-6296
[b] https://orcid.org/0000-0002-5038-9785
[c] https://orcid.org/0000-0002-8193-0597

Talpini, J., Civiero, N., Sartori, F. and Savi, M.
A Federated Approach to Enhance Calibration of Distributed ML-Based Intrusion Detection Systems.
DOI: 10.5220/0013376600003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 840-848
ISBN: 978-989-758-737-5; ISSN: 2184-433X

ing paradigm in the context of network intrusion detection to address the issues recalled above (Campos et al., 2021; Talpini et al., 2023). It embodies the principles of *focused collection* and *data minimization*, and can especially mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning, including high communication efficiency and low-latency data processing (Kairouz et al., 2021).

In addition, it is fundamental to note that network intrusion detection is a safety-critical application, where the consequences of wrong predictions (i.e., attack yes/no, or which kind of attack) can be severe and costly. In fact, a network administrator would require a model that not only is *accurate*, but also *trustworthy*, i.e., able to indicate when its predictions are likely to be incorrect. In literature, some works have already pointed out the strong need for models that are *trustworthy* (or *reliable*) in the considered domain (Talpini et al., 2024).

A fundamental aspect for enhancing the trustworthiness of a ML model relies on its *calibration* property. A model is said to be calibrated if the probability associated with the predicted class label matches the empirical frequency of its ground truth correctness (Guo et al., 2017). As a concrete example, let us consider an Internet Service Provider that wants to offer IDS service to its customers and suppose that the system relies on a ML model to identify whether network traffic belongs to an attack or not. It would be beneficial to ensure that the model is calibrated, so that, out of a certain number of predictions with a given confidence level, say 90%, we expect roughly the 90% of samples to be correctly classified. Ensuring such property makes it possible to understand to what extent the ML model can be trusted in its classification activity.

However, calibrating a model is not always an easy task. A model is typically calibrated *after* its training by using a *calibration dataset* (Böken, 2021). While this is not an issue in centralized settings, where a lot of users' data is made available for model training, and part could also be used for calibration, it becomes unfeasible in privacy-preserving scenarios – such as those targeted by FL – where data cannot be shared and must be kept local. On the other hand, performing calibration locally on the globally-trained model by means of FL may be only partially effective.

To tackle this problem we designed a novel *federated calibration* module based on a well-known calibration method, i.e., *Platt scaling* (Böken, 2021). Our strategy embraces the federated learning principles, and it can be applied to any given pre-trained model. Given its peculiarities, it is especially suitable in federated settings where also the calibration operation, and not only the learning, exploits data from the local clients. Especially, this module can operate in federated scenarios without requiring modifications to the standard *FedAvg* (McMahan et al., 2017) model aggregation procedure and without the need of changing or retraining the base model.

We evaluated the effectiveness of our approach using the ToN-IoT dataset (Sarhan et al., 2021), a widely adopted benchmark for studying attacks on IoT infrastructures (Alsaedi et al., 2020), focusing on how data heterogeneity affects the calibration properties of the proposed approach. For comparison, we included a baseline classifier trained in a federated learning framework, which demonstrated poor calibration properties, as well as a *centralized calibration* approach where the calibration module is trained using data retrieved from clients (i.e., IoT devices) at a centralized location. Additionally, we tested a personalized calibration approach where each client independently trains its own calibrator (i.e., *local calibration*). Our results show that calibrating the model is effective to enhance its reliability and that our federated calibration approach provides *(i)* better calibration performance than local calibration, and *(ii)* only slightly worse performance than centralized calibration, which however breaks privacy-preserving constraints. In addition, this is obtained without significantly altering the model's classification performance in terms of F1-Score and accuracy.

To summarize, our main contributions are:

- The definition of a federated calibration module that works with any pre-trained classifier, but that is best suited to a federated learning environment.
- An evaluation of the proposed calibration approach in IoT intrusion detection scenarios with different clients' data heterogeneity, and its comparison to other relevant baselines.

The structure of the paper is organized as follows. Section 2 introduces the related work, while Section 3 introduces the proposed federated calibration approach. Section 4 describes the experimental setup and the dataset utilized. In Section 5 we provide and discuss the numerical results, and conclude the paper by highlighting the main takeaways and lessons learned in Section 6.

2 RELATED WORK

Intrusion Detection Systems are traditionally regarded as a secondary layer of defense, designed to monitor network traffic and identify malicious activ-

ities that primary security measures (e.g., firewalls) are not able to identify (Moustafa et al., 2018). IDSs are typically categorized as either *signature-based* or *anomaly-based* (Tauscher et al., 2021). Signature-based IDSs, also known as misuse detection systems, use pattern recognition techniques to compare current network traffic against known attack signatures. In contrast, anomaly-based IDSs build a model of normal network behavior and identify any deviations from this baseline as potential intrusions. However, they often suffer from a high false positive rate (Al-Garadi et al., 2020).

In this paper, we focus on signature-based intrusion detection, with a specific focus on IoT environments, and in the following subsection we report on the relevant work related to data-driven systems, based on machine learning, in this context. Later, we also recall relevant recent strategies that have been proposed to improve the calibration of a model in a federated setting.

2.1 ML-Based Intrusion Detection Systems

In recent years, data-driven approaches for developing IDSs have received a lot of attention from the research community (Shone et al., 2018; Liu and Lang, 2019; Tauscher et al., 2021) considering different methods such as random forests, support vector machines, neural networks or clustering techniques. In particular, machine learning and deep learning are emerging as powerful data-driven approaches capable of learning and identifying malicious patterns in network traffic, making them highly effective for detecting security threats in networks (Shone et al., 2018), and in particular in IoT environments (Chaabouni et al., 2019; Sarhan et al., 2022).

However, the majority of intrusion detection approaches proposed for the IoT domain rely on centralized architectures, where IoT devices transmit their local datasets to cloud datacenters or centralized servers. This setup leverages the substantial computing power of these centralized systems for model training (Campos et al., 2021). As a result, the federated learning paradigm emerged as a promising alternative to traditional centralized approaches in this domain, and it is possible to find a few examples in the literature exploring its feasibility (Rahman et al., 2020; Campos et al., 2021; Aouedi et al., 2022; Rey et al., 2022; Talpini et al., 2023).

For instance, in (Rey et al., 2022) the authors propose a FL-based framework to detect malware, based on both supervised and unsupervised models. More precisely, the authors perform a binary classification with a multi-layer perceptron and an autoencoder on balanced datasets, which present the same class proportions for every client. Another contribution is represented by the paper (Campos et al., 2021), in which the authors investigated a FL approach on a realistic IoT dataset and showed the issues related to the so-called statistical heterogeneity, arising from the fact that typically different devices experience different kinds of attacks.

Another relevant contribution in the realm of reliable models for intrusion detection is (Talpini et al., 2024), where the authors show how uncertainty quantification can enhance the trustworthiness of an IDS, with a focus on the usual centralized setting. Our paper is aligned with the vision of (Talpini et al., 2024), as we propose a federated approach to obtain and deploy calibrated models, which is a primary requirement for an uncertainty-aware model.

2.2 Model Calibration in Federated Settings

Calibration has been extensively studied in the literature within the traditional centralized setting (see (Guo et al., 2017) for a comprehensive overview), as machine learning models, including deep neural networks, are often poorly calibrated. Approaches to address calibration issues can either involve modifications to the model itself or adjustments to the standard loss function (e.g., temperature scaling, focal loss), or they can be applied post-training, assuming the availability of a fresh dataset independent of the training set. In line with the latter approach, several methods exist to improve calibration, including *isotonic regression* and *Platt scaling* (Guo et al., 2017).

However, the calibration of machine learning models in federated settings has received limited attention so far. Among the few relevant efforts, recent research (Peng et al., 2024) demonstrated that existing federated learning schemes often fail to ensure proper model calibration following weight aggregation, raising concerns about the deployment of FL models in safety-critical domains. To address this, they proposed a post-hoc calibrator based on a multi-layer perceptron with a substantial number of free parameters, which introduces a risk of overfitting and leads to a non-negligible communication overhead between the clients and the central server.

Similarly, (Qi et al., 2023) explored the problem of model calibration in FL by proposing a strategy that leverages prototype-based summaries of each client's data cluster to facilitate effective calibration. However, their approach is designed for a cross-silo FL setting, which permits peer-to-peer communica-

tion between clients. In contrast, this work focuses on the cross-device FL scenario, where clients communicate exclusively with a central server.

Another recent study (Chu et al., 2024) proposed modifying the local training loss by incorporating a calibration loss to encourage better model calibration. While effective for neural network models, this method is unsuitable when pre-trained classifiers are involved, as envisioned in this paper, since it requires access to the training process.

All the previous works are related to model calibration without focusing on any specific application domain. To the best of our knowledge, in the domain of FL-based network intrusion detection, the issue of proper model calibration remains unexplored, and this paper tries to fill this gap.

3 OUR APPROACH TO FEDERATED CALIBRATION

As mentioned earlier, proper calibration is a prerequisite for reliable learning performance. While model trustworthiness is typically studied in centralized settings, here we propose a *federated calibration module* (or *federated calibrator*) that can take as input any pre-trained model. It operates in federated scenarios without requiring modifications to the standard FedAvg (McMahan et al., 2017) aggregation architecture. More specifically, the module adopts a novel approach: not only is the model trained in a federated way, but it is also calibrated in the same manner, leveraging local calibration information from the local clients.

The desired goal of the calibrator module is as follows: given any 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 achieve this goal, we chose to rely on *Platt scaling* – also known as sigmoid or logistic calibration – due to its simplicity and parametric nature that naturally fits the federated learning workflow.

More specifically, suppose a *binary classification* problem and 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 by means of a *sigmoid function* with two learnable parameters, denoted here as a and b, as follows:

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

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 of Equation (1) can be easily extended to a *multi-class classification* task by applying the same equation to each class independently, and then normalizing the output to ensure a valid probability interpretation. The great advantage of Platt scaling is that it has only a few parameters (as it scales linearly with the number of classes) making it suitable even for calibration datasets of small size. Additionally, it is versatile and, as already pointed out, it can be applied to any pre-trained model.

Our simple yet effective proposal is to *train this calibrator following the standard federated learning pipeline*, e.g. by adopting FedAvg (McMahan et al., 2017) in a centralized server. Specifically, the server (i.e., FL model aggregator) sends to the clients (or to a random subset of them) a pre-defined calibrator model (in terms of parameters a and b of the sigmoid function) for local training. Each client trains the given model with its own data, by minimizing a given loss function (i.e., cross-entropy) and hence computing a local update of the weights, here denoted as $\boldsymbol{w} \equiv [a, b]$ following Equation (1).

At the end of each round, the server collects all local updates related to a and b and combines them to update the central model parameters. Following this iterative process, an arbitrary number of clients can concur to model calibration without transferring

Algorithm 1: Federated Calibrator.

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

the collected data (i.e., calibration dataset) to a centralized location, since only locally-trained sigmoid function parameters need to be sent. As said, the common aggregation function adopted in this paper for federated calibration is FedAvg, which computes the global updated weights $\boldsymbol{w}_{\text{global}}$ as a weighted average over all clients' weights $\boldsymbol{w}_k$.

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

Overall, the proposed calibration workflow is coordinated by the central server. It takes as input a pre-trained *base model* and provides, as output, a calibrated version of it by executing the federated calibration module and the related workflow, whose details are reported in Algorithm 1.

Note that the base model may have been trained following any possible training paradigm. However, given its peculiarities, our calibration procedure is especially recommended in the case of a model trained by means of federated learning, as it follows the same workflow. Specifically, the model could first be trained using FL, and then calibrated giving it as input to our federated calibration module.

4 DATASET AND EXPERIMENTAL SETUP

4.1 Dataset Description

For exploring the feasibility of our approach, the choice of a realistic dataset plays a crucial role. In (Campos et al., 2021) it is possible to find an extensive review of different existing datasets related to intrusion detection in the IoT domain. As done in that paper and given its properties, here we exploit the *NF-ToN-IoT dataset* (Sarhan et al., 2021) *version 2*, which is based on the ToN_IoT set (Alsaedi et al., 2020). Every row of the dataset represents a network flow characterized by 43 features, and each flow is labeled as belonging to one over a total of 10 classes, including benign traffic and nine different attacks. In the dataset there are approximately 17 million rows with 63.99% representing attack samples and 36.01% representing benign samples.

To distribute data among clients we exploited a common partition method used in the literature, which is based on a symmetrical Dirichlet distribution, governed by a concentration parameter α. Such a parameter can be used to indicate the *statistical heterogeneity*, where lower values of α (say < 1) lead to more heterogeneous settings (i.e., more different sample distributions across classes for the different clients). In particular, we considered 10 clients with varying α between 0.2 and 0.9. It is worth mentioning that in a network intrusion detection scenario it is common to have statistical heterogeneity (Campos et al., 2021) and therefore we concentrated our analysis on a heterogeneous scenario, i.e., with $\alpha < 1$.

Moreover, to leverage the simplicity of the calibration module, which allows training the calibration with a small amount of data, we applied a random sub-sampling of the classes except for the minority one, so that all the classes are equally represented at calibration time.

4.2 Adopted Classifier and Calibrator Implementation

As a base model, we considered an extreme gradient-boosted decision tree classifier implemented in the library *XGBoost* (Chen and Guestrin, 2016). This model often outperforms more complex models (e.g., deep learning models) on tabular data such as the one considered here (Shwartz-Ziv and Armon, 2022; Markovic et al., 2022), being very appealing in the case of network traffic classification.

Each client locally trains its own XGBoost model, then the central server aggregates the local models to define a single global model following the federated learning schema. Boosted decision trees are aggregated as follows: at each FL round, each client sends its boosted tree to a central server; the server then treats these trees as bootstrapped versions of a classifier and combines them into an ensemble to create the final model. So, 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 (Beutel et al., 2020), and the federated training of the three-based model required 50 training rounds and 10 local epochs per round.

Regarding the calibration module, Platt scaling has been implemented through Pytorch (Paszke et al., 2019) as a fully-connected neural network without hidden layers, with a diagonal weight matrix and with a sigmoid (i.e., the Platt scaling sigmoid of Equation (1)) as an activation function. This allowed an easy implementation of the calibration module directly in the Flower workflow. The calibrator's FL training rounds took 20 rounds and 10 local epochs per round.

4.3 Baseline Strategies

To evaluate the effectiveness of our proposed approach, we considered the following baselines:

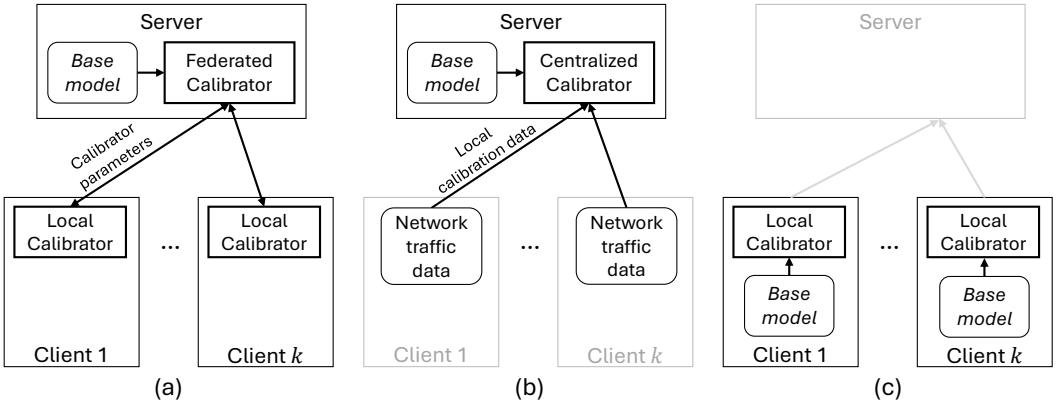

Figure 1: Architectural view of our federated calibration approach (a), centralized calibration (b) and local calibration (c).

- **Base model**. This is the classifier obtained in a federated way, as specified in Section 4.2, without any calibration module.
- **Centralized calibration**. It represents the most favorable scenario in terms of calibration performance, where the central server has access to a calibration dataset so that calibration can be performed in the usual centralized setting. However, this is not a privacy-preserving strategy as a portion of clients' data needs to be shared with the central server.
- **Local calibration**. In this strategy, each client independently trains its own calibrator using its own local data. In this way, the calibration is personalized with respect to each client.
- **Proposed approach**. This is the federated calibration approach proposed in Section 3.

A comparison of the different calibration approaches is shown in Fig. 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 baselines and across each value of α, to ensure a fair comparison of the models' performance.

5 PERFORMANCE EVALUATION

The main goal of this paper is to improve the calibration of a given classifier. To assess how much a model is calibrated, a commonly adopted metric is the so-called *Expected Calibration Error* (ECE) (Guo et al., 2017). To compute it, the predicted probability is divided into a set of bins and the discrepancy between the empirical probability (i.e., the fraction of correctly-classified samples belonging to the given confidence range) and the predicted probability

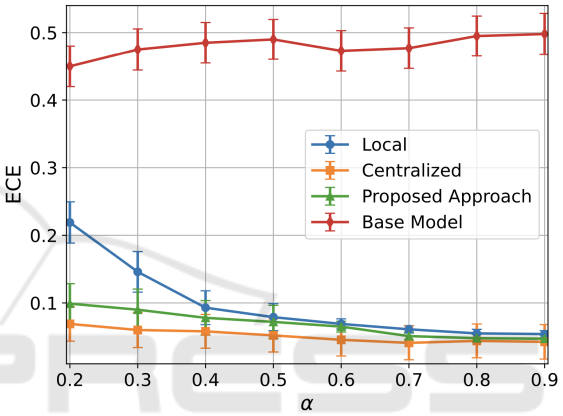

Figure 2: Expected Calibration Error for the considered models as a function of the clients' data statistical heterogeneity. The curves represent the mean ECE along with the 95% confidence intervals, calculated over 20 experiments with different random seeds.

is evaluated. More precisely, it is defined as:

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

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 lie in the range $[0,1]$, with lower values indicating a better calibrated model.

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.

Figure 2 shows the ECE as a function of the heterogeneity parameter α for the considered baselines. It can be seen that the base model is highly uncalibrated for all the values of α, thus raising concerns

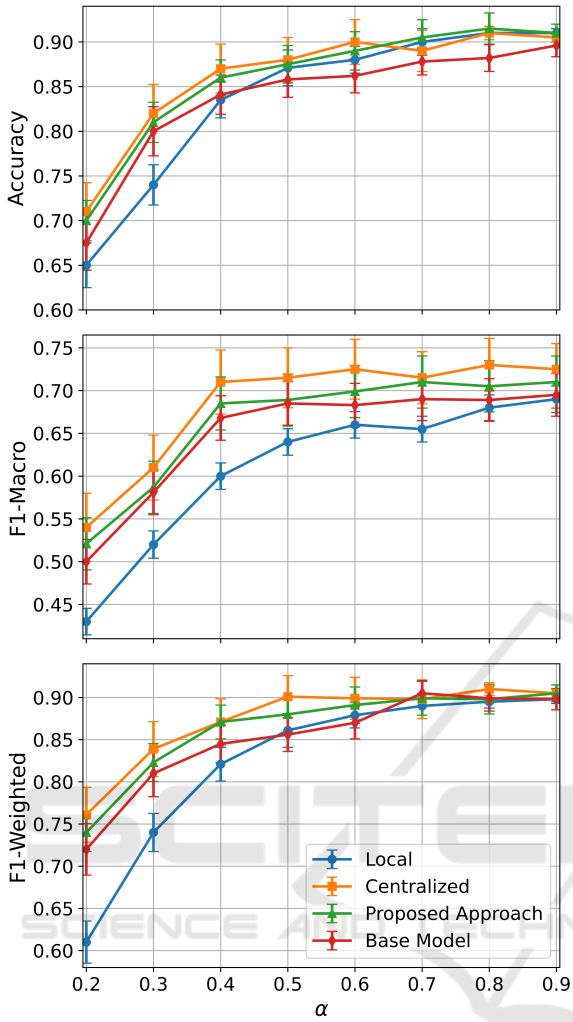

Figure 3: Accuracy and F1-Scores for the considered models as a function of the clients' data statistical heterogeneity. The curves represent the mean values of each metric along with the 95% confidence interval, computed over 20 different experiments with varying random seeds.

about the reliability of standard FL-based models. On the other hand, we can see that the proposed approach shows a comparable behavior to centralized calibration, but without the need for a centralized calibration dataset, thus guaranteeing privacy and efficiency since data remains local.

Moreover, the proposed approach leads to better calibration with respect to a personalized, local calibration. The difference, as expected, is higher in more heterogeneous settings, where local data are not representative of the overall population as typically happens in a network intrusion detection scenario. In fact, for instance, certain regions may be more exposed to certain cyber attacks than others, leading to imbalances in the data collected from each client. The

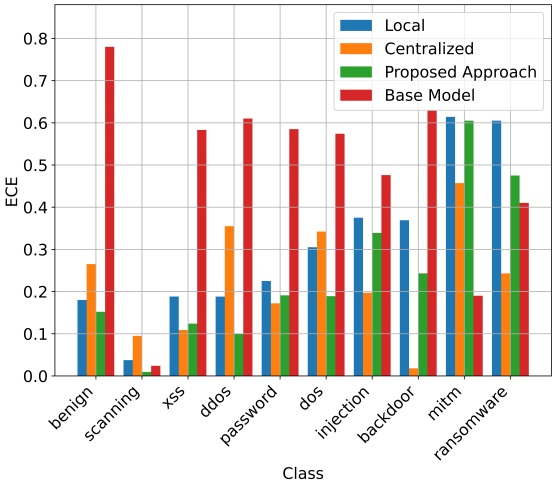

Figure 4: Per-class ECE for the evaluated models with fixed data heterogeneity ($\alpha = 0.3$). The classes are ordered in decreasing order based on their sample size. The ECE represents a mean value, computed over 20 different experiments with varying random seeds.

ability of the model to perform well under these conditions is important to ensure its effectiveness in practical, large-scale intrusion detection systems.

It is worth noting that the calibration module does not negatively impact the classification performance, as shown in Figure 3, for accuracy, F1-Macro and F1-Weighted. Here, we can see that our approach guarantees similar classification performance to the centralized calibration case and to the base model. Again, local calibration shows a sub-optimal behavior, especially in more heterogeneous settings. In addition, similar results are obtained for the Weighted F1-score, with even closer performance between our approach, centralized calibration and base model.

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. The results highlight that minority classes (e.g., ransomwere, mitm, etc.) exhibit significantly higher ECE values, indicating a stronger calibration challenge in these cases for all the baselines, showing how Platt scaling struggles in performing its task.

6 CONCLUSION

In this paper, we proposed a novel federated calibration module relying on a calibrator that can be trained following the standard federated learning pipeline. The proposed module demonstrates good calibration performance at small α values, i.e., with heteroge-

neous client's data, and maintains comparable performance to server-based centralized calibration as α increases (i.e., $\alpha > 0.4$), while offering advantages in terms of privacy, as no calibration data must be shared with the server. At the same time, our federated calibration outperforms a local calibration strategy, where each client calibrates separately the base model (e.g. trained by means of FL): separate calibration steps at different nodes might leverage partially representative data and, hence, result in non-trustworthy models.

In addition, the classification performance in terms of accuracy and F1-score of our proposed approach is not affected with respect to the base model, and it is comparable to that of a model obtained by performing centralized calibration.

These features make our approach particularly well-suited for the deployment of reliable ML-based Intrusion Detection Systems, where data are typically unbalanced and where privacy and efficient resource usage are essential. However, it still faces challenges to work well with under-represented classes, highlighting an area for potential improvements. In addition, for future works, we plan to investigate other approaches to calibration in a federated learning setting, like isotonic regression and/or methods based on the conformal prediction framework.

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

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., and Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys Tutorials*, 22(3):1646–1685.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., and Anwar, A. (2020). TON-IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access*, 8:165130–165150.

Aouedi, O., Piamrat, K., Muller, G., and Singh, K. (2022). Intrusion detection for Softwarized Networks with Semi-supervised Federated Learning. In *ICC 2022*.

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. (2020). Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390*.

Böken, B. (2021). On the Appropriateness of Platt Scaling in Classifier Calibration. *Information Systems*, 95:101641.

Campos, E. M., Saura, P. F., González-Vidal, A., Hernández-Ramos, J. L., and et al. (2021). Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges. *Computer Networks*, page 108661.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., and Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys Tutorials*, 21(3):2671–2701.

Chen, T. and Guestrin, C. (2016). Xgboost: A Scalable Tree Boosting System. In *KDD 2016*.

Chu, Y.-W., Han, D.-J., Hosseinalipour, S., and Brinton, C. (2024). Unlocking the Potential of Model Calibration in Federated Learning. *arXiv preprint arXiv:2409.04901*.

European Union Agency for Cybersecurity (2023). ENISA Threat Landscape 2023. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023. [Accessed: 09-December-2024].

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. In *ICML 2017*.

Hassija, V., Chamola, V., Saxena, V., Jain, D., and et al. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access*, 7:82721–82743.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., et al. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity*, 2(1):1–22.

Liu, H. and Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*, 9:4396.

Markovic, T., Leon, M., Buffoni, D., and Punnekkat, S. (2022). Random Forest based on Federated Learning for Intrusion Detection. In *AIAI 2022*. Springer.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pages 1273–1282.

Mothukuri, V., Parizi, R., Pouriyeh, S., Huang, Y., and et al. (2020). A survey on Security and Privacy of Federated Learning. *Future Generation Computer Systems*.

Moustafa, N., Hu, J., and Slay, J. (2018). A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey. *Journal of Network and Computer Applications*, 128.

Paszke, A. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS 2019*.

Peng, H., Yu, H., Tang, X., and Li, X. (2024). FedCal: Achieving Local and Global Calibration in Federated Learning via Aggregated Parameterized Scaler. In *ICML 2024*.

Qi, Z., Meng, L., Chen, Z., Hu, H., Lin, H., and Meng, X. (2023). Cross-silo Prototypical Calibration for Federated Learning with Non-iid Data. In *Multimedia 2023*.

Rahman, S. A., Tout, H., Talhi, C., and Mourad, A. (2020). Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning? *IEEE Network*, 34(6):310–317.

Rey, V., Sánchez, P. M. S., Celdrán, A. H., and Bovet, G. (2022). Federated Learning for Malware Detection in IoT devices. *Computer Networks*, 204:108693.

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., and Khan, M. (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science*, 171:1251–1260.

Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., and Portmann, M. (2022). Feature Extraction for Machine Learning-based Intrusion Detection in IoT Networks. *Digital Communications and Networks*.

Sarhan, M., Layeghy, S., and Portmann, M. (2021). Evaluating Standard Feature Sets Towards Increased Generalisability and Explainability of ML-based Network Intrusion Detection. *arXiv preprint arXiv:2104.07183*.

Shone, N., Ngoc, T. N., Phai, V. D., and Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50.

Shwartz-Ziv, R. and Armon, A. (2022). Tabular Data: Deep Learning is not All You Need. *Information Fusion*, 81:84–90.

Talpini, J., Sartori, F., and Savi, M. (2023). A Clustering Strategy for Enhanced FL-Based Intrusion Detection in IoT Networks. In *ICAART 2023*.

Talpini, J., Sartori, F., and Savi, M. (2024). Enhancing Trustworthiness in ML-Based Network Intrusion Detection with Uncertainty Quantification. *Journal of Reliable Intelligent Environments*, pages 1–20.

Tauscher, Z., Jiang, Y., Zhang, K., Wang, J., and Song, H. (2021). Learning to Detect: A Data-driven Approach for Network Intrusion Detection. In *IPCCC 2021*.

Tsimenidids, S., Lagkas, T., and Rantos, K. (2022). Deep Learning in IoT Intrusion Detection. *Journal of Network and Systems Management*, 30.