Federated Learning for DDoS Attack Detection: Enhancing Security with Misclassification Penalties

Aymen Ben Said, JingTao Yao, Nashid Shahriar
Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada
{Aymen.BenSaid, Jingtao.Yao, Nashid.Shahriar}@uregina.ca

***Abstract*—Anomaly detection is the task of identifying abnormal behavior and unusual patterns. In the context of telecommunications, anomaly detection plays an important role in maintaining network security by identifying security threats such as cyberattacks, security breaches, and network performance issues. There are two potential actions to take when detecting anomalies in the network: either blocking the suspicious traffic or sending it for further analysis. Although traditional machine learning methods using centralized models have shown promising results for anomaly detection, they come with the data privacy limitation since they require the dataset to be processed on a central server. This limitation motivates the use of decentralized approaches that preserve data privacy. Recently, Federated Learning (FL) has been introduced to tackle this challenge. FL is a distributed machine learning method that enables clients to collaboratively train a global model by aggregating model updates while keeping the local data unexposed, which guarantees that sensitive private information remains secure and protected. In this context, we adopt the FL model for anomaly detection. Furthermore, we propose three model aggregation methods namely; FedAvgFCR, FedProxFCR, and FedNovaFCR as extensions of FedAvg, FedProx, and FedNova, respectively. The proposed variants penalize clients during training based on the misclassified testing instances from the previous round's global model. The experimental results of our proposed variants using five subsets of a Distributed Denial of Service (DDoS) attack dataset reveal promising results in terms of average local and global performance metrics compared to FedAvg, FedProx, and FedNova.**

***Index Terms*—Federated Learning, Machine Learning, Decentralized Learning, Anomaly Detection**

I. Introduction

Machine Learning (ML) plays a major role in many industry applications due to its ability to learn hidden patterns and associations among historical data, and solve challenging problems. ML methods usually work by analyzing large datasets and make predictions based on the implicitly learned patterns. Anomaly detection has gained great attention among ML applications for its critical role in many areas [1]–[3]. The main goal of anomaly detection in cybersecurity is to identify outliers from the usual behavior that may indicate potential problems such as cyberattacks, security breaches, and network performance issues. However, a major limitation of traditional ML methods in this context is that they usually involve storing user's data on a central server for training, which may raise privacy and security concerns, especially when dealing with sensitive data. To overcome this challenge, Federated Learning (FL) has been introduced. FL is a distributed ML framework proposed to tackle data privacy concerns [4]. FL allows different models to be trained locally on client devices without exposing user's local data to the central server. The coefficients of these locally trained models are then sent to the central server and aggregated into a global model. The global model is consequently distributed back to the client devices, and the same process is repeated for multiple rounds to improve the global performance. While FL offers security and privacy advantages, it usually comes with a relatively lower performance as a trade-off compared to the standard centralized ML methods [5]. Thus, our objective in this paper is to tackle this limitation by proposing three new FL aggregation variants namely FedAvgFCR, FedProxFCR, and FedNovaFCR. These variants are built on top of the existing aggregation strategies, FedAvg [6], FedProx [7], and FedNova [8], respectively. This is achieved by introducing an additional term as part of the loss function. The added term penalizes the participating clients based on their misclassified instances during local training. To evaluate the performance of our proposed model aggregation variants, we have conducted several experiments using subsets of a Distributed Denial of Service (DDoS) attack dataset. To our knowledge, these existing methods have not been previously applied to DDoS attack detection. The experimental results are very promising as they demonstrate considerable improvements in local and global average performance throughout the individual FL rounds, as well as enhanced final models for the majority of experiments.

II. Related work

ML has proven to be an effective method for supervised, unsupervised, and reinforcement learning problems, including anomaly detection. Chaudhary et al. introduced a Graph Neural Network (GNN) approach to detect anomalous data points in social networks, relying on deep learning techniques to analyze the topological features related to social connection graphs [9]. Shojafar et al. proposed an LSTM-based approach to identify anomalies in packet forward control protocol (PFCP) signaling traffic to improve the security and reliability of 5G networks [10]. Similarly to ML, FL has been explored in the literature for many domains. Nguyen et al. introduced DÏoT which is a self-learning system that detects compromised devices in IoT networks [11]. Salam et al. introduced an FL-based model for credit card fraud detection in the financial sector [12].

One of the main differences in FL settings is the aggregation step. In FedAvg, each client trains its local model using its

private data, with the same learning rate and number of epochs, and then sends the model coefficients to the central server that averages them to update the global model [6]. FedProx is an extension of FedAvg that is proposed to address the challenge of non-IID (non-independent and identically distributed) data that may result in poor convergence of the global model. FedProx introduces a proximal term to the local objective function to prevent local model training from deviating significantly from the global model during the FL rounds. The proximal term is controlled by the hyperparameter μ to determines how much the local updates are restricted to stay close to the global model. The modified local objective function for a client k is defined as follows: $\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2$, where w represents the local model coefficients, w^t denotes the global model parameters at round t, $F_k(w)$ depicts the local loss function for client k, and $\frac{\mu}{2}\|w - w^t\|^2$ is the proximal term [7]. Unlike FedProx, FedNova was proposed to deal with system heterogeneity. FedNova assumes that the participating clients train their local models with varying numbers of epochs given their computational power, thus, the model updates are averaged by introducing an extra normalization step according to the number of epochs each client performs at each round [8]. In addition to FedAvg and FedProx, we also employ FedNova in our experimentation. SCAFOLD is another FL aggregation strategy that addresses data' statistical heterogeneity and local training inconsistency to prevent client drifts. SCAFOLD uses control variates on the client and server sides that adjust the global model updates during local training instead of just averaging them. The SCAFOLD objective function for each client k is defined as follows: $\nabla F_k(w) + c - c_k$, where $\nabla F_k(w)$ represents the gradient of the local objective function of client k, c is the global control variate, and c_k is the local control variate for client k [13]. MOON relies on a contrastive loss term that encourages the current local model to be aligned closer to the global model received at the beginning of the current round while also discouraging it from being similar to the local model of the previous round [14]. Zeno is another FL aggregation method that relies on a preference scoring mechanism to rank the participating clients according to their contributions in reducing the loss of the global model. In doing so, Zeno assumes that there may be many faulty clients as long as at least one client is reliable [15].

III. Proposed Model Aggregation Variants

Although researchers have proposed various approaches to address the challenges related to data and system heterogeneity that can lead to performance issues, those approaches still require further attention. We introduce three model aggregation variants as extensions to FedAvg, FedProx, and FedNova. These variants work by adding a weighted penalty term to the objective function during FL training rounds. Assume that round t results in an aggregated model w_t. After distributing this model to the participating clients for training in round $t + 1$, it is evaluated locally on the independent testing set associated with each client. The classification metrics obtained are then used to compute the weighted penalties and incorporated into the objective functions during the clients' training. The objective functions of our proposed FedAvgFCR and FedProxFCR variants are depicted in Equations 1 and 2, respectively, where $\lambda\ FCR_k(w^t)$ is the additional weighted penalty term that corresponds to the false classification rates, λ is a hyperparameter that controls the strength of the penalty.

$$\min_w h_k(w; w^t) = F_k(w) \cdot (1 + \lambda\ FCR_k(w^t)) \tag{1}$$

$$\min_w h_k(w; w^t) = F_k(w) \cdot (1 + \lambda\ FCR_k(w^t)) + \frac{\mu}{2}\|w - w^t\|^2 \tag{2}$$

Note that FedNovaFCR uses the same local objective function as FedAvgFCR, however, the main difference is in the aggregation step at the server level. FedNovaFCR computes a normalized update vector for each client k as follows $\Delta w_k = W_{r-1} - w_k$, and tracks the number of local optimization steps τ_k. The server then aggregates these updates as depicted in Equation 3 to ensure that the contribution of each client is scaled according to its optimization effort.

$$W_r = W_{r-1} - \frac{1}{|K|} \sum_{k \in K} \frac{\Delta w_k}{\tau_k} \tag{3}$$

The pseudo-algorithms of our FedAvgFCR, FedProxFCR, and FedNovaFCR variants are given in Algorithms 1, 2, 3, 4, and 5.

Algorithm 1 FedAvgFCR

Input: Clients training sets D, clients testing sets T, number of rounds R, number of clients K, number of epochs E, learning rate η
Output: Aggregated global model W_R
Initialize global model W_0 (server level)
for r from 1 to R **do**
  for each client $k \in K$ locally **do**
    $\lambda_k \leftarrow$ Evaluate(W_{r-1}, T_k)
    $w_k \leftarrow$ FedAvgFCRTraining(W_{r-1}, λ_k)
  end for
  Global model aggregation (server level):
  $W_r \leftarrow \frac{1}{|K|} \sum_{k \in K} w_k$
end for
return Final aggregated global model W_R

Algorithm 2 FedAvgFCRTraining

Initialize local model $w_k \leftarrow W_{r-1}$
for e from 1 to E **do**
  for each batch $b \in D_k$ **do**
    $L(w_k) = MSE(w_k) \cdot (1 + \lambda_k)$
    $w_k \leftarrow w_k - \eta \nabla L(w_k)$
  end for
end for
return updated model w_k

Algorithm 3 FedProxFCRTraining

Initialize local model $w_k \leftarrow W_{r-1}$
for e from 1 to E **do**
  for each batch $b \in D_k$ **do**
    $L' = MSE(w_k) \cdot (1 + \lambda_k)$
    $L(w_k) = L' + \frac{\mu}{2}\|w_k - W_{r-1}\|^2$
    $w_k \leftarrow w_k - \eta \nabla L(w_k)$
  end for
end for
return updated model w_k

Note: FedProxFCR's pseudo-algorithm is the same as FedAvgFCR with an additional proximal term μ as input.

Algorithm 4 FedNovaFCR

Input: Clients training sets D, clients testing sets T, number of rounds R, number of clients K, number of epochs E, learning rate η
Output: Aggregated global model W_R
Initialize global model W_0 (server level)
for r from 1 to R **do**
  for each client $k \in K$ locally **do**
    $\lambda_k \leftarrow \text{Evaluate}(W_{r-1}, T_k)$
    $\Delta w_k, \tau_k \leftarrow \text{FedNovaFCRTraining}(W_{r-1}, \lambda_k)$
  end for
  Global model aggregation (server level):
  $W_r \leftarrow W_{r-1} - \frac{1}{|K|} \sum_{k \in K} \frac{\Delta w_k}{\tau_k}$
end for
return Final aggregated global model W_R

Algorithm 5 FedNovaFCRTraining

Initialize local model $w_k \leftarrow W_{r-1}$
Initialize local steps $\tau_k \leftarrow 0$
for e from 1 to E **do**
  for each batch $b \in D_k$ **do**
    $L(w_k) = MSE(w_k) \cdot (1 + \lambda_k)$
    $w_k \leftarrow w_k - \eta \nabla L(w_k)$
    $\tau_k \leftarrow \tau_k + 1$
  end for
end for
$\Delta w_k \leftarrow W_{r-1} - w_k$
return model difference Δw_k, number of local steps τ_k

IV. Experimentation

In our FL architecture, we rely on autoencoder models. In scope of our anomaly detection problem, the autoencoders are trained on benign traffic data to learn the patterns associated with normal behavior, and the anomalies are then identified based on a reconstruction error (we set this threshold to 0.2 in scope of our experimentation). We use autoencoder models with an input size of 41, and two hidden layers containing 128 and 64 neurons, respectively. The latent space that encodes the data features consists of size 32 neurons. The decoder reconstructs the input data by expanding the latent space representation through the two hidden layers in reverse to produce a reconstructed 41-dimensional vector. We use stochastic gradient descent with adaptive learning rate algorithms to optimize the learning performance.

At each FL round, the individual client models are tested using their respective labeled testing sets that consist of both benign and attack traffic to evaluate their performance in both scenarios before aggregation. The aggregated global model is also tested locally using the same testing sets to obtain the classification metrics (specific to each client), which are then used to construct the additional weighted penalty term for training in the subsequent round. Note that the Mean Squared Error (MSE) is employed as the base objective function for all aggregation methods. We conducted five separate experiments using a DDoS attack dataset [16], divided fairly among the five experiments. The dataset includes benign and attack traffic, and it suits our FL study as it provides enough data to test DDoS detection in a realistic scenario. For each experiment, each client is trained using its own unique dataset of dimensions: 41 features and 162257 samples, and tested on a unique test set with 84000 samples. Each experiment consists of 20 rounds. Every client is trained for 10 epochs with a batch size of 32, except for FedNova and FedNovaFCR, which use a varied number of epochs and batch sizes. The learning rate was set to 0.001 for all methods. Furthermore, early stopping is used to prevent client models from continuing training if no improvement is observed after five consecutive epochs. Concerning the hyperparameters, we set $\mu = 0.01$ for FedProx and FedProxFCR, and $\lambda = 0.1$ for all our proposed variants. The experiments were conducted on a PC with the specifications: Intel® Core™ i7-10510U CPU @ 1.80GHz 2.30 GHz, and 16.0 GB RAM.

A. Results and Discussion

Our experimentation results are reported in terms of local and global performance recorded throughout the 20 training rounds. The local average performance refers to the mean performance metrics of the four client models before the aggregation step, with each model evaluated on its corresponding client's test set. The global average performance represents the mean performance metrics of the global model obtained after aggregation, evaluated using the independent test sets of the four clients. Fig. 1 visualizes the average local and global accuracies, precisions, and F1 scores of the four clients per round before and after the model aggregation step using FedAvg. Similarly, Fig. 3 visualizes the results related to FedProx and Fig. 5 presents the corresponding results for FedNova. While FedAvg and FedNova outperform FedProx in all performance metrics, FedProx exhibits a more stable convergence of both the local and global models.

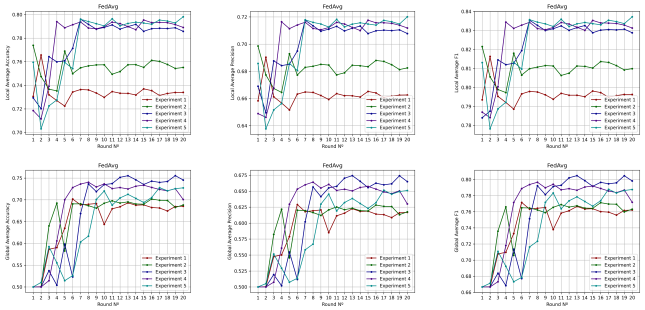

Fig. 1. FedAvg Local and Global Performance

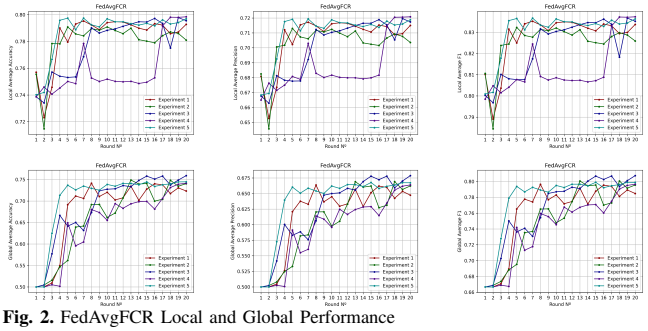

Fig. 2. FedAvgFCR Local and Global Performance

The local and global results of our proposed model aggregation variants FedAvgFCR, FedProxFCR, and FedNovaFCR are illustrated in Figs. 2, 4, and 6, respectively. TABLES. I and II present the local and global performance in the final

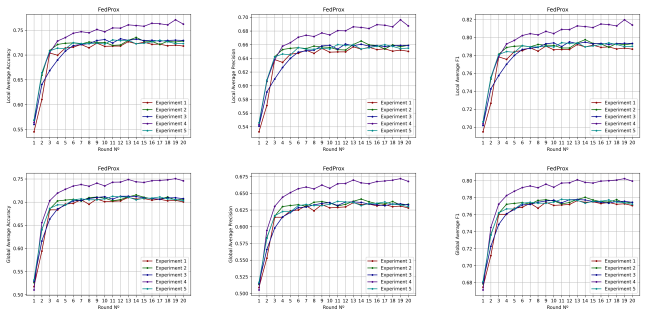

Fig. 3. FedProx Local and Global Performance

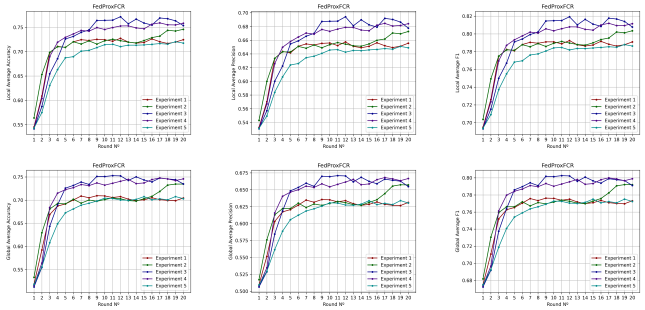

Fig. 4. FedProxFCR Local and Global Performance

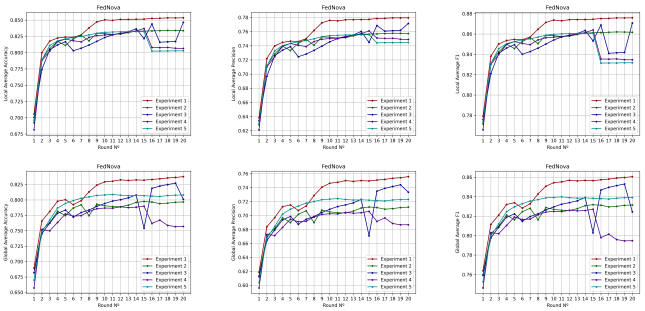

Fig. 5. FedNova Local and Global Performance

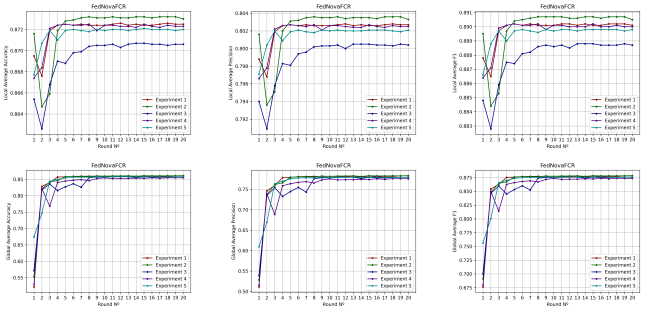

Fig. 6. FedNovaFCR Local and Global Performance

round for all methods. The experimental results are very promising, as they demonstrate improved performance in favor of our proposed variants throughout the FL rounds, as well as improved final local and global models in the majority of the experiments. In addition to the final round's local and global results, TABLE. III presents the lowest and highest local and global performance results recorded in the final round across all experiments, regardless of the experiment number. Fig. 7 visualize these results. Although FedProxFCR showed few improvements over FedProx, it should be noted that the difference is minimal (almost negligible), with some improvements observed in favor of FedProxFCR in the global minimum results. FedAvgFCR and FedNovaFCR consistently outperformed their counterparts in every metric. The local results from FedAvgFCR reveal a more stable convergence compared to FedAvg, except for experiment 4. Although experiment 4 shows lower convergence stability, the final local model in this experiment outperforms its FedAvg counterpart in all performance metrics, as shown in TABLE. I. The local results in TABLE. I further demonstrate the superiority of FedAvgFCR over FedAvg across all performance metrics in the rest of the experiments, with a very minor exception in experiment 5. The global results for FedAvgFCR also show better convergence over the 20 rounds. TABLE. II shows that FedAvgFCR outperforms FedAvg across all performance metrics in all five experiments. The lowest final global model's accuracy was recorded at 72.33%, outperforming its counterpart that achieved 68.53% (both results recorded in the first experiment). The highest final global model's accuracy of FedAvgFCR reached 75.90%, which surpasses the 74.55% accuracy achieved by FedAvg in the third experiment. FedProxFCR has also shown encouraging results compared to FedProx. FedProxFCR consistently maintained the convergence stability associated with FedProx while delivering improved local and global performance across all rounds. In addition, FedProxFCR showed better results in the final models, with a slight exception in the final local results recorded in experiments 4 and 5, and the global results observed in experiments 4. As shown in TABLE. III and Fig. 7, the lowest final global model accuracy for FedProxFCR was 70.32%, observed in experiment 5, which exceeds that of FedProx, which recorded 70.03% in experiment 1. The highest final global model accuracy of 74.60% was achieved in experiment 4, resulting in a tie between the two methods. FedNovaFCR outperformed FedNova both locally and globally in all performance metrics, in all rounds, including the final round, as depicted in TABLES. I and II. It also showed improved local and global convergence stability compared to FedNova, as illustrated in Fig. 6. As demonstrated in TABLE. III and Fig. 7, the lowest final global model accuracy achieved by FedNovaFCR was 85.49%, which surpasses FedNova's lowest accuracy of 75.69%, both recorded in experiment 4. In Experiment 2, FedNovaFCR achieved the highest final global model accuracy of 86.14%, which outperforms the 83.77% accuracy recorded by FedNova in Experiment 1.

V. Conclusion and Future Work

We have proposed three model aggregation variants for FL in anomaly detection as extensions to the well-known FedAvg, FedProx, and FedNova methods, namely FedAvgFCR, FedProxFCR, and FedNovaFCR. Our proposed variants introduces a weighted penalty during clients' training, determined by the misclassified instances. The experimental results against their base methods using a DDoS attack datasets demonstrate

TABLE I. Final Round's Local Performance

	FedAvg	FedAvgFCR	FedProx	FedProxFCR	FedNova	FedNovaFCR
			Experiment 1			
Accuracy	0.734	**0.7928**	0.7181	**0.7246**	0.8535	**0.8725**
Precision	0.6626	**0.715**	0.6503	**0.6557**	0.7795	**0.8027**
F1-score	0.7963	**0.8337**	0.787	**0.7909**	0.8759	**0.8901**
			Experiment 2			
Accuracy	0.7551	**0.781**	0.7282	**0.7459**	0.8337	**0.873**
Precision	0.6825	**0.7036**	0.659	**0.6728**	0.7573	**0.8033**
F1-score	0.8099	**0.8259**	0.7931	**0.8035**	0.8617	**0.8905**
			Experiment 3			
Accuracy	0.7858	**0.7955**	0.7294	**0.7535**	0.8466	**0.8706**
Precision	0.7078	**0.7174**	0.6587	**0.6777**	0.7716	**0.8004**
F1-score	0.8288	**0.8352**	0.7933	**0.8077**	0.8708	**0.8887**
			Experiment 4			
Accuracy	0.7891	**0.7984**	**0.7623**	0.7585	0.8065	**0.8723**
Precision	0.7119	**0.7208**	**0.6874**	0.6842	0.7489	**0.8025**
F1-score	0.8313	**0.8373**	**0.8139**	0.8115	0.8346	**0.89**
			Experiment 5			
Accuracy	**0.7981**	0.7969	**0.723**	0.7176	0.8028	**0.872**
Precision	**0.7201**	0.7188	**0.655**	0.6489	0.7447	**0.8021**
F1-score	**0.8372**	0.8363	**0.7902**	0.7863	0.8317	**0.8898**

TABLE II. Final Round's Global Performance

	FedAvg	FedAvgFCR	FedProx	FedProxFCR	FedNova	FedNovaFCR
			Experiment 1			
Accuracy	0.6853	**0.7233**	0.7003	**0.7045**	0.8377	**0.8583**
Precision	0.6169	**0.6473**	0.6281	**0.6313**	0.7557	**0.7801**
F1-score	0.7621	**0.7848**	0.7707	**0.7732**	0.8606	**0.8762**
			Experiment 2			
Accuracy	0.6881	**0.7404**	0.7064	**0.7347**	0.7967	**0.8614**
Precision	0.6176	**0.6621**	0.6336	**0.6572**	0.712	**0.7836**
F1-score	0.7631	**0.7956**	0.7746	**0.792**	0.8315	**0.8785**
			Experiment 3			
Accuracy	0.7455	**0.759**	0.7074	**0.735**	0.8009	**0.8579**
Precision	0.665	**0.6786**	0.6327	**0.6548**	0.7334	**0.7794**
F1-score	0.7981	**0.8074**	0.7745	**0.7911**	0.8244	**0.8758**
			Experiment 4			
Accuracy	0.7008	**0.7413**	**0.746**	**0.746**	0.7569	**0.8549**
Precision	0.63	**0.6637**	**0.6679**	0.6667	0.6868	**0.776**
F1-score	0.7717	**0.7965**	**0.7995**	0.799	0.7948	**0.8736**
			Experiment 5			
Accuracy	0.7275	**0.7447**	0.7029	**0.7032**	0.8081	**0.8584**
Precision	0.651	**0.6672**	0.6301	**0.6304**	0.7232	**0.7801**
F1-score	0.7875	**0.7989**	0.7722	**0.7724**	0.8392	**0.8762**

TABLE III. Final Round's Local-Global Min-Max Performance for all methods and experiments (Ei denotes Experiment i)

	FedAvg	FedAvgFCR	FedProx	FedProxFCR	FedNova	FedNovaFCR
Local Min Accuracy	0.734-E1	**0.781-E2**	**0.7181-E1**	0.7176-E5	0.8028-E5	**0.8706-E3**
Local Max Accuracy	0.7981-E5	**0.7984-E4**	**0.7623-E4**	0.7585-E4	0.8535-E1	**0.873-E2**
Global Min Accuracy	0.6853-E1	**0.7233-E1**	0.7003-E1	**0.7032-E5**	0.7569-E4	**0.8549-E4**
Global Max Accuracy	0.7455-E3	**0.759-E3**	**0.746-E4**	**0.746-E4**	0.8377-E1	**0.8614-E2**
Local Min Precision	0.6626-E1	**0.7036-E2**	**0.6503-E1**	0.6489-E5	0.7447-E5	**0.8004-E3**
Local Max Precision	0.7201-E5	**0.7208-E4**	**0.6874-E4**	0.6842-E4	0.7795-E1	**0.8033-E2**
Global Min Precision	0.6169-E1	**0.6473-E1**	0.6281-E1	**0.6304-E5**	0.6868-E4	**0.776-E4**
Global Max Precision	0.665-E3	**0.6786-E3**	**0.6679-E4**	0.6667-E4	0.7557-E1	**0.7836-E2**
Local Min F1-score	0.7963-E1	**0.8259-E2**	**0.787-E1**	0.7863-E5	0.8317-E5	**0.8887-E3**
Local Max F1-score	0.8372-E5	**0.8373-E4**	**0.8139-E4**	0.8115-E4	0.8759-E1	**0.8905-E2**
Global Min F1-score	0.7621-E1	**0.7848-E1**	0.7707-E1	**0.7724-E5**	0.7948-E4	**0.8736-E4**
Global Max F1-score	0.7981-E3	**0.8074-E3**	**0.7995-E4**	0.799-E4	0.8606-E1	**0.8785-E2**

promising performance in terms of average local and global performance metrics. Given this fact, we plan to apply our variants to the intrusion detection problem and explore their integration with other model aggregation methods. Furthermore, we plan to explore the dynamic reweighting of our penalty term by tuning the hyperparameter λ to incentivize better training for clients based on the 3-way decision model [17].

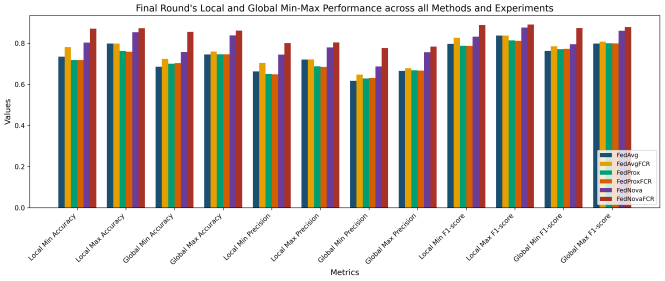

Fig. 7. Final Round's Local-Global Min-Max Performance for all methods and experiments

REFERENCES

[1] A. Chouhan, N. Shahriar, and J.T. Yao, "Hcl: A hybrid cnn-lstm framework for intrusion detection in sdn-iot networks," in *ICNC'25*, pp. 254–258, 2025.

[2] S. Natha, "A systematic review of anomaly detection using machine and deep learning techniques," *Quaid-e-Awam University Research Journal of Engineering, Science & Technology*, pp. 83–94, 2022.

[3] A. Ben Said, E. A. Mohammed, and M. Mouhoub, "An implicit learning approach for solving the nurse scheduling problem," in *Neural Information Processing*, pp. 145–157, 2021.

[4] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 1–23, 11 2022.

[5] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," *The Second Workshop on Distributed Infrastructures for Deep Learning*, pp. 1–8, 2018.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, pp. 429–450, 2020.

[8] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 7611–7623, 2020.

[9] A. Chaudhary, H. Mittal, and A. Arora, "Anomaly detection using graph neural networks," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*, pp. 346–350, 2019.

[10] R. Pell, M. Shojafar, and S. Moschoyiannis, "Lstm-based anomaly detection of pfcp signaling attacks in 5G networks," *IEEE Consumer Electronics Magazine*, vol. 14, no. 1, pp. 56–64, 2025.

[11] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏot: A federated self-learning anomaly detection system for iot," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 756–767, 2019.

[12] M. Abdul Salam, K. Fouad, D. Elbably, and S. Elsayed, "Federated learning model for credit card fraud detection with data balancing techniques," *Neural Computing and Applications*, pp. 6231–6256, 2024.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*, pp. 5132–5143, 2019.

[14] Q. Li, B. He, and D. Song, " Model-Contrastive Federated Learning ," in *Conference on Computer Vision and Pattern Recognition*, pp. 10 708–10 717, 2021.

[15] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6893–6901, 2019.

[16] M. S. Khan, B. Farzaneh, N. Shahriar, and M. M. Hasan, "Dos/ddos attack dataset of 5G network slicing," 2023, https://dx.doi.org/10.21227/32k1-dr12.

[17] J.T. Yao, C. Cornelis, G. Wang, and Y. Yao, "Uncertainty and three-way decision in data science," *International Journal of Approximate Reasoning*, vol. 162, pp. no. 109 024, 2023.