Federated-Learning-Assisted Failure-Cause Identification in Microwave Networks

Tara Tandel[1], Omran Ayoub[2], Francesco Musumeci[1], Claudio Passera[3], Massimo Tornatore[1]
[1]Politecnico di Milano, Italy. [2]University of Applied Sciences and Arts of Southern Switzerland, Switzerland
[3] SIAE Microelectronics, Italy

***Abstract*—Machine Learning (ML) adoption for automated failure management is becoming pervasive in today's communication networks. However, ML-based failure management typically requires that monitoring data is exchanged between network devices, where data is collected, and centralized locations, e.g., servers in data centers, where data is processed. ML algorithms in this centralized location are then trained to learn mappings between collected data and desired outputs, e.g., whether a failure exists, its cause, location, etc. This paradigm poses several challenges to network operators in terms of privacy as well as in terms of computational and communication resource usage, as a massive amount of sensible failure data is transmitted over the network. To overcome such limitations, Federated Learning (FL) can be adopted, which consists of training multiple distributed ML models at multiple decentralized locations (called "clients") using a limited amount of locally-collected data, and of sharing these trained models to a centralized location (called "server"), where these models are aggregated and shared again with clients. FL reduces data exchange between clients and a server and improves algorithms' performance thanks to sharing knowledge among different domains (i.e., clients), leveraging different sources of local information in a collaborative environment. In this paper, we focus on applying FL to perform failure-cause identification in microwave networks. The problem is modeled as a multi-class ML classification problem with six pre-defined failure causes. Specifically, using real failure data from an operational microwave network composed of more than 10000 microwave links, we emulate a multi-operator scenario in which one operator has partial knowledge of failure causes during the training phase. Thanks to knowledge sharing, numerical results show that FL achieves up to 72% precision in identifying an unknown particular class concerning traditional ML (non-FL) approaches where training is performed without knowledge sharing.**

***Index Terms*—Microwave networks, failure identification, root-cause analysis, machine learning, federated learning, data privacy**

I. Introduction

Machine Learning (ML) is transforming failure management in communication networks by leveraging monitoring data to automate vital failure-management operations, such as failure detection, identification, and localization. However, traditional ML techniques for failure management require gathering data on one central server. Even if performance might be convincing, the data transfer and process volume are enormous, making it harder to transmit and preserve data in a centralized location. Furthermore, data transmission may expose privacy threats, and as well the central server may be attacked and release sensitive data. In this regard, Federated Learning (FL) is a promising candidate to enable distributed data training and assist in developing high-performance models shared across many parties and hence preserve data privacy [1]. FL has already been widely adopted in several contexts where data owners are concerned about privacy, e.g., in the health sector, where patients' data consists of sensitive information that needs to be protected to comply with legal, administrative or ethical constraints.

In this study, we focus on the application of FL for failure-cause identification in microwave networks, assuming that different operators cooperate in a privacy-preserving manner, i.e., without sharing detailed information regarding the failures in their networks. Following the FL approach, failure data can be gathered and stored in the network equipment of each operator, where independent training is performed by each operator using local data. Then, the operators can collaboratively share their models and allow a central server (e.g., a coordinator) to aggregate them into a unified model that can be re-distributed to and adopted by all the operators.

More specifically, we consider a scenario with three microwave network operators aiming to discriminate between six failure causes. We assume that one operator (namely, *Partial Knowledge Operator (PKO)*) has no historical knowledge (i.e., no training data) of one of the failures at a time. Hence, exploiting knowledge from the other operators and using FL, we aim to evaluate under which conditions, e.g., failure cause, the PKO can improve its classification accuracy and precision. To validate the effectiveness of the FL-assisted failure-cause identification, we compare its classification accuracy and precision against two benchmark scenarios, where i) all the operators know all failure causes and perform independent training without sharing any data or ML model, and ii) available data from all the operators are used by a central server that performs training with global data.

This article is organized as follows. In Section II relevant existing work is discussed. Section III provides background information on Federated Learning. Section IV formally defines the problem addressed in this paper. In Section V the proposed FL-assisted failure-cause identification framework is discussed. Finally, we provide numerical results in Section VI and draw paper conclusion in Section VII.

II. Related Work

ML-assisted failure network management continues to attract the attention of academic and industrial research communities due to the proven capabilities of ML in solving a

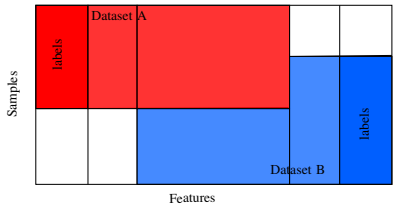


Fig. 1: Illustration of HFL where the overlapping features from data samples held by different participants are taken to train a model jointly [18].

wide range of failure management problems [2]–[4]. Most of the works focusing on ML-assisted network management, however, consider centralized ML schemes [5]–[10]. For instance, Ref. [5] investigates the problem of microwave link failure detection using a Long/Short-Term Memory (LSTM)-based feature fusion network. In our previous work [6], we tackle the problem of failure-cause identification in microwave networks in a semi-supervised fashion, and in Ref. [7] we exploit explainable artificial intelligence to explain ML-based models for failure-cause identification. Moreover, Refs. [9] and [10] focus on detecting link anomalies, while Ref. [3] investigates failure detection in cellular data networks.

The application of FL for network management has started to receive attention only recently [11]–[15]. Moreover, the existing literature is very limited and is mainly focused on the application of FL to preserve the data privacy of different data owners in a multi-operator (or several data owners) network scenario. For failure network management, Ref. [16] proposes an FL-based solution that enables operators to collaboratively develop an ML model for quality-of-transmission estimation in optical networks while preserving data privacy. Similarly, Ref. [12] demonstrates the benefits of an FL framework among different operators and data owners while preserving data privacy. With respect to these works, our work leverages the application of FL to improve the knowledge of one operator that suffers from an incomplete data set (has no historical knowledge of one of the failures) by exploiting the knowledge of other operators. To the best of our knowledge, no prior works exploited FL to evaluate under which conditions one operator (or data owner) can improve its knowledge by exploiting that of other data owners in the learning federation.

III. Background on Federated Learning

FL is an approach based on collaborative ML among different players, called *clients*, that cooperates by performing ML-model training using local data. In this paradigm, a centralized *server* (or coordinator) distributes an initial model to local data owners, who train a model using their respective local dataset and then share trained models with the server. The centralized server is in charge of aggregating the different models obtained from the clients and then sharing a unique aggregated model. We call each exchange of data between clients and the server a round. This way, data owners do not expose their data to a central server or any other data owner in such a context [17].

A. Types of FL based on the distribution of the data

Two primary forms of FL exist, Horizontal and Vertical FL (HFL and VFL, respectively). The form of FL adopted indicates how data is distributed between the clients. Fig. 1 shows an FL setting where the x-axis shows the feature space of the samples of each client. We can see an overlapping area where clients share overlapping data features. However, the y-axis shows the client's samples' space where all the sample points are; it depicts that the samples differ in most parts. In HFL, data features are aligned across the participants, but they differ in data samples.

Different from HFL, VFL applies to the case where the participants share overlapping data samples, i.e., the data samples are aligned amongst the participants, but they differ in data features. For example, two banks can store data related to a shared set of clients, but different information (i.e., different sets of features) is available for each client. It resembles the situation where data is vertically partitioned inside a tabular view [18], [19]. In our work, we refer to HFL as we consider a scenario in which different microwave network operators represent clients of the FL architecture. Since operators do not share links and usually the features of the links are the same in every working site, we can say that they have no overlapping samples yet several (if not all) similar features.

B. FL Algorithms

The three main flavors of FL algorithms are: Federated Stochastic Gradient Descent (FedSGD), Federated Learning with Dynamic Regularization (FedDyn), and Federated averaging (FedAvg) [20]. FedSGD uses gradient descent, a first-order iterative optimization algorithm, to find a local minimum of a differentiable function. The idea is to take repeated steps in the opposite direction of the gradient of the current point to find the local minimum. In stochastic gradient descent, gradients are computed on a random subset of the dataset and are then used to make one step of the gradient descent [21]. The algorithm converges when the update to the gradient at each step is small enough.

FedDyn is concerned with minimizing communication, allowing for significantly more processing and optimization at the client level. Specifically, the model dynamically modifies the client's objective with a penalty term in each round. When model parameters converge, they do so to stationary points of the global empirical loss.

Finally, FedAvg is a generalization of FedSGD that allows local nodes to perform more than one batch update on local data and exchanges the updated weights rather than the gradients. In our work, we use FedAvg, which employs more mini-batches than FedSGD, resulting in reduced communication overhead. Note that FedDyn can be used for heterogeneous datasets, which are not considered in our work. Three essential parameters govern the amount of computation in FedAvg: i) ρ, the fraction of clients that perform computation during each round; ii) S, the number of training steps each client performs; and iii) M, the mini-batch size used for client updates. During each round, FedAvg picks a ρ-fraction of participants and

computes the weights across all of the players' data [22]. FedAvg optimizes the finite-sum objective:

$$\min_{w \in \mathbb{R}^d} f(w), \quad f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w) \tag{1}$$

where w is a vector that contains d model parameters. In supervised learning, we treat the function $f_i(w)$ as loss function $f_i(w) = l(x_i, y_i; w)$, where an input-output pair (x_i, y_i) is one of n given labeled examples, often referred to as training examples. The objective function $f_i(w)$ is defined by the model parameters w conditioned on n labeled examples. The problem can thus be interpreted as finding the value of w which minimizes the average loss over all n training examples.

In an FL setting, data is distributed among k clients, with their respective data $P_0, P_1, ...P_k$. The number of training examples held by client k is denoted by $n_k = |P_k|$. Each client then holds a part P_k of all training examples and computes $F_K(w)$, the average loss on client k, and then we can rewrite the objective function as a weighted sum over all $F_k(w)$:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(W), \text{ where } F_k(w) := \frac{1}{n_k} \sum_{i \in P_k} f_i(w) \tag{2}$$

Distributing the data and computational burden leads us to re-formulate the objective function $f(w)$ from Eq. 1 to 2 [23].

IV. Problem Settings and Data Description

A. Problem Statement

We model failure identification as a multi-class FL-based classification problem to identify the failure root cause with distributed datasets. The classifier is trained collaboratively between several clients, with the features described in Sec. IV-C model. We adopted FL to emulate a scheme with N clients (Network Operators) holding some certain amount of data but with one of the clients missing a failure type (we refer to this client as Partial Knowledge Operator (PKO)). These N clients aim to build a robust classifier without sharing data and help the other operators recognize the missing failure type.

B. Failure Causes

The structure of a microwave communication link includes a microwave radio at the transmitter site, connected to a directional antenna via a transmission line. Because of physical restrictions and changes in the propagation environment, the signal quality of the microwave may fluctuate over time. These variations may cause network breakdown and make the network unavailable. To measure the unavailability of a microwave link, the "UnAvailability Seconds (UAS)," measure is defined, which accounts for the number of seconds when the system is unavailable in a given observation interval [24]. The UAS value in the considered observation interval is computed as the sum of all the time intervals containing at least ten consecutive severely errored seconds.

Six distinct failure root causes are considered in our work.

1) *Deep Fading* consists of a significant rise in channel attenuation, resulting in a significant loss in signal-to-noise ratio. Heavy rain, snow, or fog are all possible causes, resulting in the multipath and shadowing effects.

2) *Extra Attenuation* occurs when received power falls considerably below a certain power threshold. Many factors can cause this problem, including route blockage (due to permanent barriers), antenna misalignment, and mounting/screwing difficulties.

3) *Interference* takes place when a receiving antenna receives numerous bitstreams owing to the overlap of other transmissions at the same frequency used by the receiver, or unexpected reflections from other links, leading it to fail to differentiate the bitstream meant for it.

4) *Low Margin* occurs when the connection is misconfigured due to a human mistake, resulting in UAS incidents. As a countermeasure, distant human interaction is necessary to configure the link's parameters appropriately.

5) *Self-Interference* happens when the connection generates local signal reflections and spurious signals that are carried to the receiver's radio component owing to hardware deterioration, resulting in random UAS on the link.

6) *Hardware failure* consists of either a temporary or permanent breakdown of equipment countermeasure. Human action is required on-site to replace the hardware equipment causing the failure.

C. Input Data Description

Our raw data comprises measurements from an Italian microwave network with 10841 radio links. For each data sample, we consider the power measurements in the 15-minute window affected by the failure, in conjunction with information on their evolution in the previous 30-minutes windows. In total, we use 35 features, grouped as follows (please refer to [6] for further details):

Design information $(x_1 \div x_3)$: G.828 performance measures ES and SES for the three 15-minutes slots

Propagation Measures $[x_4 \div x_{35}]$: The performance and power measure sampled during the last 15-minutes windows in which there is at least a UAS.

A human expert labeled each data sample, indicating one of the failures causes described earlier in Section IV-B.

V. Methodology

This section describes our methodology. We assume to have three operators (referred to as clients), each with a subset of the data. Also, we have a central server in charge of receiving and averaging the received weights from clients. Fig. 2 shows the overall framework for failure identification using FL, and it consists of three main steps. First, we split the data among the clients. Second, we define the stopping conditions and run the hyper-parameter search for the FedAvg algorithm. Finally, we train and test the model obtained using the FedAvg algorithm described in Sec. III-B. We describe the three phases in detail below.

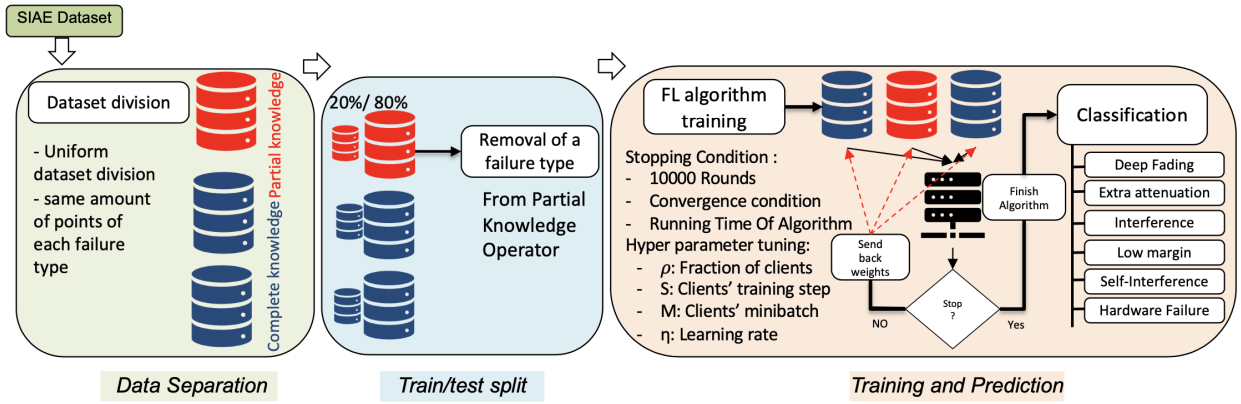


Fig. 2: Overview of the steps to taken to obtain a prediction using FedAvg algorithm.

A. Dataset separation

To emulate a realistic scenario, we split our dataset among the three operators guaranteeing that information on each microwave link is available to one operator only, i.e., we assume that the operators do not share the physical infrastructure.

Moreover, to be fair to all clients and to be able to quantify the impact of using FL (whether it degrades or improves accuracy) when a specific failure type is missing from one client, we split the data set uniformly into three sub-datasets in terms of the number of data points per failure type, as shown in Tab. I. This can happen in realistic scenarios, e.g., when an operator does not have full knowledge of all types of failure that may occur. Finally, to evaluate FL performance, for the datasets of each client, we consider an 80/20% training/testing split.

TABLE I: Dataset distribution across the different clients.

Label N.	Failure Type	Points in c1	Points in c2	Points in c3
0	Deep Fading	76	76	75
1	Extra Attenuation	155	155	154
2	Interference	13	12	12
3	Low Margin	51	50	50
4	Self-Interference	50	49	49
5	Hardware Failure	326	325	325

TABLE II: Convergence time with different conditions.

%	No. of rounds	duration (min)
0.1%	50	∞
1%	50	∞
0.1%	30	42
1%	30	35
0.1%	20	28
1%	20	20

B. Stopping condition

FedAvg can be terminated in three ways: 1) a specified maximum time has passed since running the algorithm without meeting the convergence conditions, 2) a maximum number of rounds is reached (in our work, we consider 10^4 rounds), or 3) the algorithm converges (i.e., convergence condition has been met). We experimented with several convergence conditions specified by the percentage of change of the loss function and the number of iterations that the loss function stays within a defined percentage change. The percentage of change of 0.1 within 20 rounds provided a fair compromise between the model's accuracy and the time spent, as indicated in the Table II[1]. It is worth noting that after 1000 cycles of learning, all of these prerequisites will be valid.

C. Hyperparameter Search

We chose Artificial Neural Network (ANN) as an ML algorithm, assuming all clients know the selected algorithm. With the number of hidden layers equal to 5, the number of nodes per layer is equivalent to 100 and uses Relu as an activation function.

The hyperparameters tuned for FedAvg are ρ, S, M, and the learning rate η. We will not halt the algorithm until it has completed 500 rounds.

TABLE III: Hyperparameters for the FedAvg Algorithm

hyperparameter	Range	value
ρ	$[0.1, 0.001, 0.00001, 0.0000001]$	1
S	[10, 100, 200 , 500]	500
M	[32, 64, 128]	64
η	0.00001	

D. Training Procedure

The training procedure of FedAvg is as follows: first, the central server sends the model's hyper-parameters to all three clients. Then, clients locally train the ML model with the hyper-parameters and then send a copy of the achieved weights of the ML model to the central server after completing the training procedure. After receiving all three sets of weights, the server averages them and sends the averaged weight to all three clients. This method will repeat until the stopping condition specified is fulfilled. At the end of the training procedure, all

[1]Measurements were collected using an Apple MacBook with 8 GB Memory, 2133 MHz LPDDR3 and a Quad-Core Intel Core i5 Processor at 1.4 GHz.
The ∞ indicates that the algorithm did not meet the convergence condition in the specified time

three clients will have the FL model, which they may test on their local test set.

We compare the performance of the FedAvg technique to our two benchmarks given in V-E to determine which one is most suited for categorizing a new point (i.e., a 45-minutes window inside UAS in the latest 15-minutes slot) in different use situations.

E. Benchmark

We benchmark the performance of the proposed FL approach against two baseline methods, *Local model* and *Global model*. The three approaches are described as follows:

- *FedAvg model*: The FL model was trained collaboratively among all three clients.
- *Local model*: This model trains each client with its local data, resembling the situation where sharing data between clients is not possible.
- *Global model*: We train this model on the aggregated dataset of the three clients.

We employ ANN with hyperparameters as V-C as the foundation technique and stochastic gradient descent for optimization in all three approaches. For FL, we utilized the FedAvg method with hyperparameters described in III.

VI. Numerical Results and Discussion

A. Performance evaluation for partial knowledge operator

For numerical evaluations, we consider six scenarios where the PKO lacks the knowledge of one of the failure classes during the training phase. The lack of points in the PKO's training dataset of a specific failure emulates a situation where the PKO aims to exploit FL to gain knowledge of the failure-cause class it lacks from other parties (clients two and three).

We compare the performance of *FedAvg* to that of the benchmark scenarios (*Global* and *Local*) in terms of i) precision of the class with missing data points inside PKO's train set (this metric allows us to evaluate the benefit of using *FedAvg* for detecting the class which PKO lacks knowledge of), and ii) overall model's accuracy, to quantify the impact of *FedAvg* on the overall model's performance (considering all classes of failure).

Model's precision in detecting the class of missing labels: Fig. 3 shows the model's precision in detecting the class which PKO lacks the knowledge of in the three considered models and in the six different scenarios, indicated with $L0$, $L1$, ..., $L5$, according to the label (i.e., the failure type) that we assume as missing during training phase for PKO. Note that the *Global* scenario, which is the case with global knowledge, represents an upper bound in terms of all metrics, and it ranges between 69% (for $L0$) and 98% (for $L5$).

Regarding *Local*, as expected, we can see that the precision is near zero, as PKO does not know a specific failure cause during training. In the test phase, the data points with ground truth as that of the missing label, resulting in a random classification into one of the known failure causes.

As for *FedAvg*, precision reaches 72% for the case of $L5$, which is the class with the highest number of data points in the data set. Conversely, in the case of $L2$, there is no such advantage in precision because it is the class of failure with the lowest number of data points in the dataset (only 12 data points of $L2$ are present in each client's data set). This shows that when full-knowledge operators have limited knowledge of the label PKO misses, PKO does not benefit from the FL setting. On the contrary, the PKO can benefit more from other parties when their data sets are abundant in data points of a label.

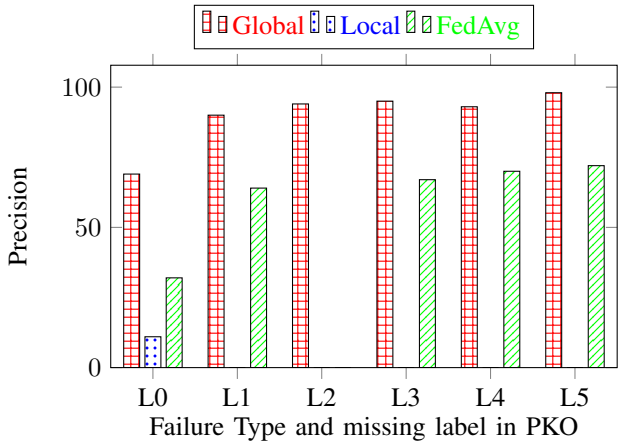


Fig. 3: Precision comparison for PKO for the three approaches for the six different scenarios.

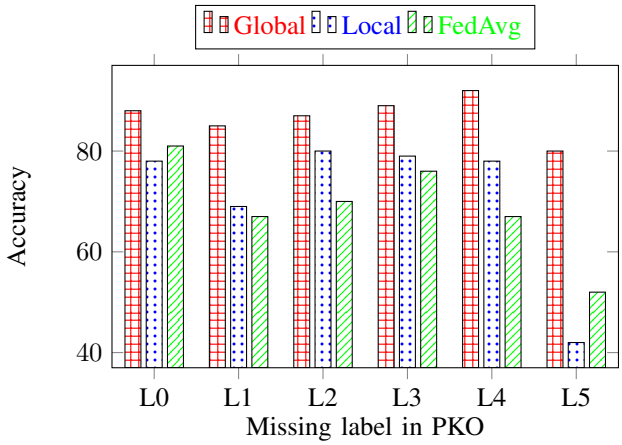


Fig. 4: Accuracy comparison for PKO for the three approaches for the six different scenarios.

Outcome 1: *FedAvg* allows for improving the precision of an unseen class of failure type, and the improvement corresponds to the knowledge other parties in FL exhibit.

Overall model's accuracy: We now compare the overall model's accuracy of the three approaches for the six scenarios. Specifically, we would like to quantify whether the improvement in precision in detecting unseen failure-cause class translates into an improvement in accuracy (and if so, to what extent) for the *Local* model.

Fig. 4 shows the accuracy of the three approaches for PKO in the six different scenarios. As expected, *Global* represents

the highest achievable accuracy (on average, *Global* has a 20% higher accuracy than the *FedAvg*). Comparing the accuracy of *FedAvg* and *Local*, we see that neither of the models outperforms the other in all scenarios. For $L0$ and $L5$, *FedAvg* shows a higher accuracy than that of *Local*, while *Local* outperforms *FedAvg* in the remaining scenarios. This shows that *FedAvg* can enhance *Local*'s accuracy only under specific conditions (in our scenario in the case when PKO lacks knowledge on deep fading ($L0$) and hardware failure ($L5$) only). In particular, for the case of Hardware Failure ($L5$), the improvement in accuracy is due to the fact that i) $L5$ class has a high number of data points (around 50% over the entire dataset) with more than 300 points in each client, and ii) Hardware Failure phenomena produce very different effects on the used features compared to the other types of failures ($L0$ to $L4$) that are all related to propagation phenomena. On the other hand, for the cases where *Local* outperforms *FedAvg*, we notice that the significant loss in accuracy is for $L2$ (interference). We relate that to the fact that the number of points categorized as $L2$ is deficient (lowest among all classes) and that the knowledge gained by PKO in the FL settings of a new label with few points seems to do more harm than good (it distracts the attention from other labels, resulting in a worse detection by *FedAvg* of other classes).

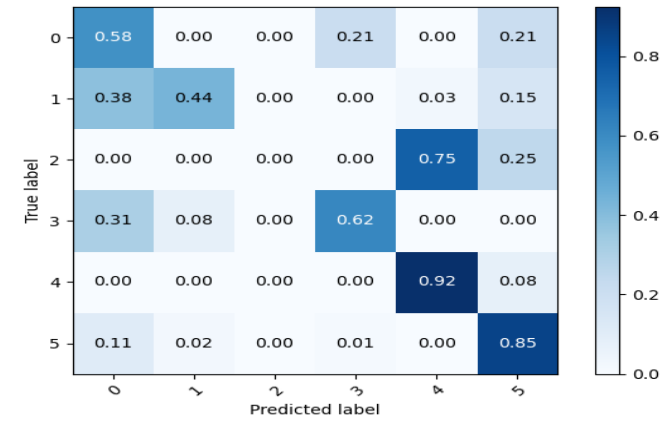
Outcome 2: *FedAvg*'s improvement in detecting the class of a failure that is unknown by the PKO does not necessarily translate into an improvement in accuracy, as *FedAvg* fails to see other failure causes as good as with the *Local* model.

Interpreting the impact of *FedAvg* on the model's accuracy: To examine in more detail the outcomes so far, we provide the confusion matrix of *FedAvg* for the scenarios when Interference ($L2$) and Hardware failure ($L5$) are absent from PKO (Fig. 5(a) and (b) respectively).

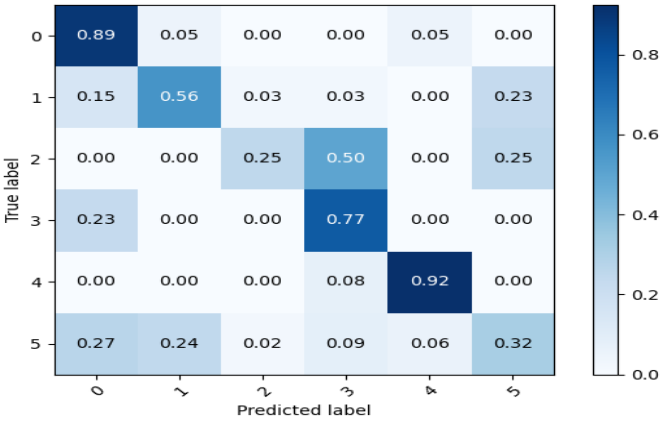
Fig. 5(a) shows that *FedAvg* was unable to categorize any points of *Interference* ($L2$). The PKO has no knowledge of interference, and there are only a few $L2$ points in full knowledge operators (each has 12 points). As a result, PKO did not benefit from full knowledge operators as intended. Fig. 5(b) shows the confusion matrix for PKO when it lacks knowledge of *Hardware failure* ($L5$). Here, we can see that PKO successfully classified 32% of the points, thanks to full knowledge operators. This further confirms how crucial the knowledge of other clients in FL can be for a partial-knowledge client.

B. Effect of FedAvg on Full Knowledge Operators

We now focus on the impact of FL on full knowledge operators, whose datasets do not miss samples from any of the labels during the training phase. Fig. 6 shows the accuracy of the three models for operators two (a) and three (b). Results show that *FedAvg* performance is comparable to that of the Local model. Results also show that full knowledge operators can achieve high accuracy (in some cases, the highest) with their local models and that they can also benefit from *FedAvg* in some cases. For instance, in the case of $L4$, we can observe



(a)



(b)

Fig. 5: Confusion matrix for the scenario where PKO is not having label two (a) and label five (b) in its train set.

a 3% improvement in accuracy in both clients compared to the *Local* model.

In some cases, this means that *FedAvg* can allow both clients to build a more robust and accurate model. Yet, the fact that PKO misses knowledge of some of the labels might impact the capability of *FedAvg*, causing the accuracy (including that of client two and client three) to drop. Finally, we note that full knowledge operators can either use *FedAvg* or rely on their *Local* model. In the latter case, the participation of full knowledge operators as a party in the FL settings can be due to other factors (e.g., economical) rather than the improvement of model performance.

VII. Conclusion

We designed an FL-based classification model using the *FedAvg* algorithm and applied it to automate the identification of six different failure causes in microwave networks. We consider a scenario with three other operators, where one operator (i.e., one "client" in FL terminology) only partially understands the failure-cause throughout the training phase. We assess the influence of knowledge sharing enabled by FL

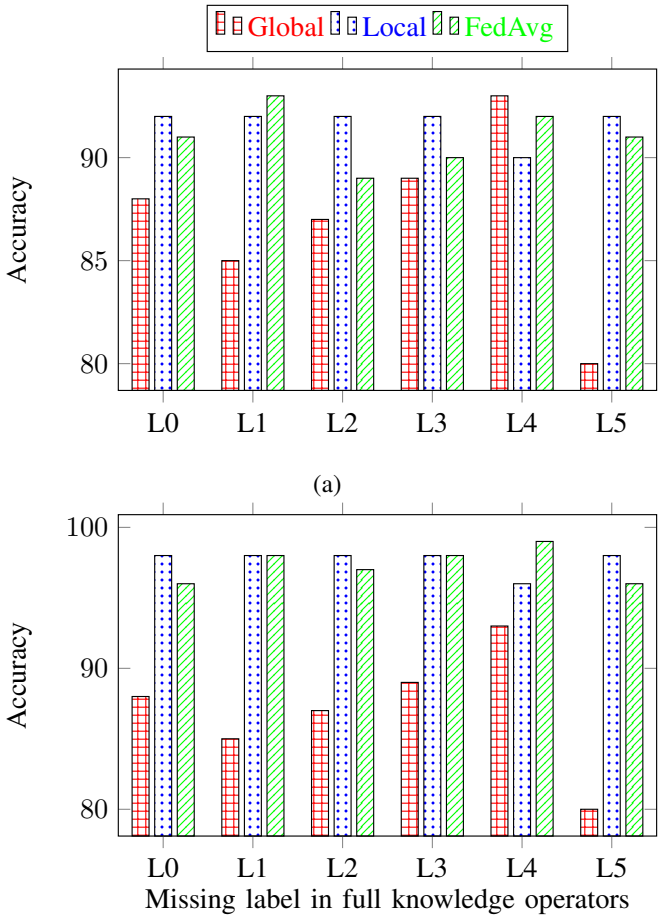


Fig. 6: Accuracy for clients two (a) and three (b)

on classification performance when a failure cause is unknown to one operator throughout the training phase. All of the models are trained using real-world data instances. Where Deep Fading and Hardware Failures' data points were lacking, we saw a gain in overall accuracy for partial knowledge operator of 3% and 10%, respectively, compared to the Local model. The accuracy of hardware failure has grown since the number of points in this class is high, which can benefit complete knowledge customers and raise the accuracy of partial knowledge clients. We also achieved a maximum of 72% for Hardware Failure. In future work, we plan to explore federated transfer learning to obtain more consistent benefits in accuracy.

REFERENCES

[1] M. Asad, A. Moustafa, and T. Ito, "Federated learning versus classical machine learning: A convergence comparison," *CoRR*, vol. abs/2107.10976, 2021.

[2] F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini, and M. Tornatore, "A tutorial on machine learning for failure management in optical networks," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4125–4139, 2019.

[3] P. Szilágyi and S. Nováczki, "An automatic detection and diagnosis framework for mobile communication systems," *IEEE transactions on Network and Service Management*, vol. 9, no. 2, pp. 184–197, 2012.

[4] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[5] Z. Ruan, S. Yang, L. Pan, X. Ma, W. Luo, and M. Grobler, "Microwave link failures prediction via lstm-based feature fusion network," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2021.

[6] F. Musumeci, L. Magni, O. Ayoub, R. Rubino, M. Capacchione, G. Rigamonti, M. Milano, C. Passera, and M. Tornatore, "Supervised and semi-supervised learning for failure identification in microwave networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1934–1945, 2021.

[7] O. Ayoub, F. Musumeci, F. Ezzeddine, C. Passera, and M. Tornatore, "On using explainable artificial intelligence for failure identification in microwave networks," in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pp. 48–55, IEEE, 2022.

[8] I. Syrigos, N. Sakellariou, S. Keranidis, and T. Korakis, "On the employment of machine learning techniques for troubleshooting wifi networks," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, 2019.

[9] L. Pan, J. Zhang, P. P. Lee, M. Kalander, J. Ye, and P. Wang, "Proactive microwave link anomaly detection in cellular data networks," *Computer Networks*, vol. 167, p. 106969, 2020.

[10] P. Casas, P. Fiadino, and A. D'Alconzo, "Machine-learning based approaches for anomaly detection and classification in cellular networks.," in *TMA*, 2016.

[11] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[12] B. Shariati, P. Safari, G. Bergk, F. I. Oertel, and J. K. Fischer, "Inter-operator machine learning model trading over acumos ai federated marketplace," in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, IEEE, 2021.

[13] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.

[14] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2021.

[15] B. Shariati, P. Safari, A. Mitrovska, N. Hashemi, J. K. Fischer, and R. Freund, "Demonstration of federated learning over edge-computing enabled metro optical networks," in *2020 European Conference on Optical Communications (ECOC)*, pp. 1–4, IEEE, 2020.

[16] N. Hashemi, P. Safari, B. Shariati, and J. K. Fischer, "Vertical federated learning for privacy-preserving ml model development in partially disaggregated networks," in *2021 European Conference on Optical Communication (ECOC)*, pp. 1–4, IEEE, 2021.

[17] P. Kairouz and et al, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[18] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, *Federated Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2019.

[19] Y. Lie, W. Yang, T. Chen, and Z. Wei, "Federated learning and transfer learning for privacy, security and confidentiality." AAAI 2019 Tutorial.

[20] R. Kontar, N. Shi, X. Yue, S. Chung, E. Byon, M. Chowdhury, J. Jin, W. Kontar, N. Masoud, M. Nouiehed, C. E. Okwudire, G. Raskutti, R. Saigal, K. Singh, and Z. Ye, "The internet of federated things (ioft): A vision for the future and in-depth survey of data-driven approaches for federated learning," *CoRR*, vol. abs/2111.05326, 2021.

[21] A. Kanekar, "Optimization and convergence of machine learning algorithms," Jun 2018.

[22] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016.

[23] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, DIDL '18, (New York, NY, USA), p. 1–8, Association for Computing Machinery, 2018.

[24] "G.828 itu-t recommendation: Error performance parameters and objectives for international, constant bit-rate synchronous digital paths," March 2000.