Multi-Output Regression for Spatial Reuse with Federated Learning

Selim F. Yilmaz, Emre Ozfatura, Kerem Ozfatura, Ozlem Yildiz and Deniz Gündüz

***Abstract*—This document describes the model we have developed for the ITU-ML5GPS-004: Federated Learning for Spatial Reuse in a Multi-BSS competition. We introduce a federated learning based model for the spatial reuse problem, where each deployment is considered a node. We employ neural network model with multi-output objective for variable number of outputs.**

***Index Terms*—spatial reuse, wireless communications, federated learning, multi-output learning**

I. Introduction

Spatial reuse operation (SR) included in the IEEE 802.11ax-2021 (11ax) amendment [1] target to increase the spectral efficiency by allowing parallel transmissions. Hence, SR operation is introduced to address the performance issues in certain scenarios where the basic service sets (BSS), which comprised of an access point (AP) and assigned STAs, are overlapping. In the case of overlapping BSSs (OBSSs), typical channel access mechanisms based on Carrier Sense Multiple Access (CSMA) underutilizes the spectral resources. To this end, by changing the sensitivity it is possible to reduce the Carrier Sense (CS) area and hence increase the frequency of channel access which, on the other hand, may lead collisions. Hence, the main objective is to design a adaptive mechanism for sensitivity of the CS that maximizes the throughput.

A. Spatial Reuse

Clear Channel Assessment CS (CCA/CS) protocol aim to ensure that any Wi-Fi device does not transmit while another is already transmitting on the same channel to avoid collusion. Hence, once a Wi-Fi device detect a transmission and decode the preamble of the transmitting device, a busy flag is raised for the channel. The Wi-Fi device can decode the preamble of the corresponding transmission if the signal strength is above certain threshold, which is often referred as the CCA/CS threshold.

SR operation target to adaptively change the sensitivity for channel sensing such that although the sensed power level is above the CCA/CS threshold, the Wi-Fi device may ignore the corresponding transmission and utilize the channel. To this end, we consider a second threshold, namely OBSS/PD threshold, which is utilized to ignore the decoded preamble of the transmission of another device and continue to occupy the channel. We remark that when there are more than one transmission, then capture affect is used that is the strongest signal is considered for CS [2]. We also want remark that SR-based opportunities are considered when inter-BSS frames are detected, in the case of detecting intra-BSS transmissions, the default CCA/CS threshold is used.

In Fig. 1, the visualization of the detection and decoding of the signal is given for IEEE 802.11 devices [2] when the devices using the same frequency and the power is fixed. AP_A can detect the signals in the gray area but only decode the red area, which is above the CS threshold. Due to the SR operation in 11ax, OBSS threshold prevents the transmission in green area to increase the utilization.

B. Federated Learning

The federated learning (FL) framework has been introduced in [3] to enable large-scale *collaborative learning* in a distributed manner and without sharing local datasets among the clients to address the privacy concerns of the end-users to some extent. Formally, FL aim to solve a problem in the following form

$$\min_{\boldsymbol{W}\in\mathbb{R}^d} f(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{\zeta\sim\mathcal{D}_i} F(\boldsymbol{W},\zeta)}_{:=f_i(\boldsymbol{W})}, \tag{1}$$

in a decentralized manner, where $\boldsymbol{w} \in \mathbb{R}^d$ denotes the model parameters, ζ is a random data sample, $\mathcal{D}_i$ denotes the dataset of client i, and F is the problem specific empirical loss function. At each iteration of FL, each client aims to minimize its local loss function $f_i(\boldsymbol{w})$ using the *stochastic gradient descent* method. Then, the clients seek a consensus on the model under supervision of the central server. The most widely used consensus strategy is averaging the local models periodically which is referred as *federated averaging (FedAvg)*. The FedAvg procedure is summarized in Algorithm 1.

To summarize the FL procedure; at the beginning of round t, a subset of the available clients $\mathcal{S}_t$ are chosen, default randomly, to participate model update. Then, each chosen client pulls the current global parameter model $\boldsymbol{w}_t$ from the PS and then performs H local updates before the consensus step, as illustrated in Algorithm 1 (lines 5-6), where

$$\mathbf{g}^{\tau}_{i,t} = \nabla_{\boldsymbol{W}} f_n(\boldsymbol{W}^{\tau-1}_{i,t}, \zeta_{i,\tau}) \tag{2}$$

is the gradient estimate of the i-th client at τ-th local iteration based on the randomly sampled local data $\zeta_{i,\tau}$.

Then, each participating user pushes its local model parameters to the central server, following H local updates, where those values are aggregated to update the parameter model, i.e.,

$$\boldsymbol{W}_{t+1} = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t} \boldsymbol{W}^{H}_{i,t}. \tag{3}$$

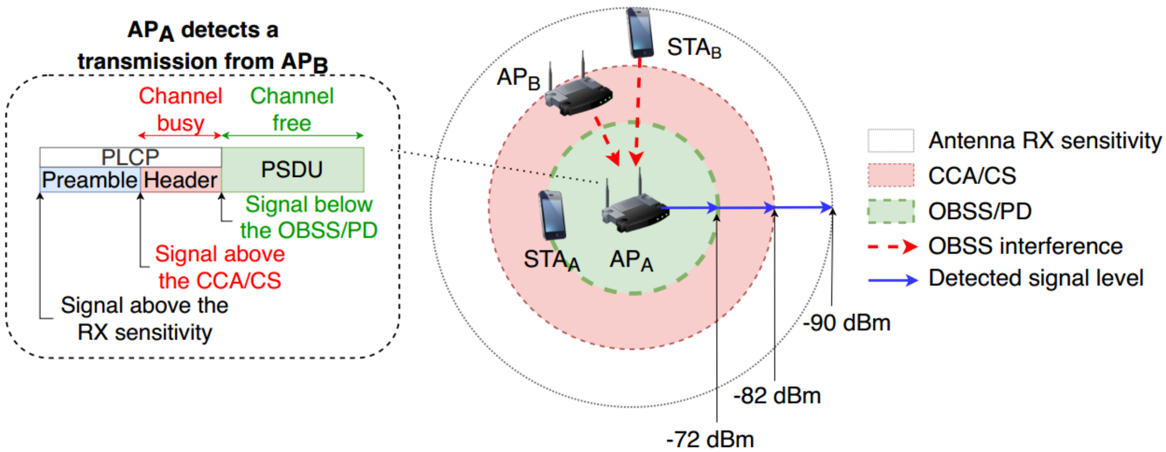

Fig. 1. Visualization of the detecting a transmission and looking for SR opportunity using CCA/CS and OBSS/PD thresholds [2].

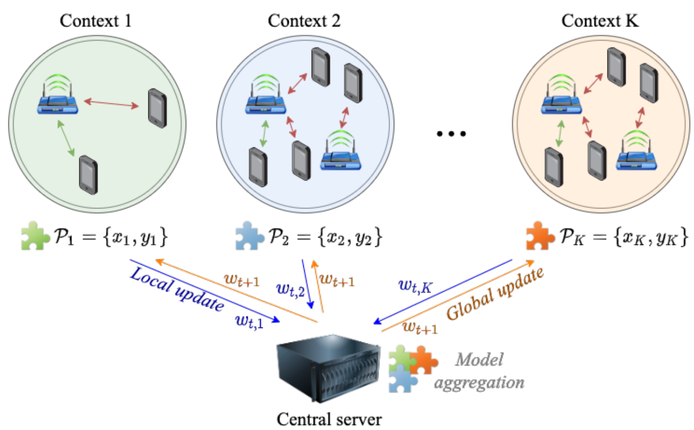

Fig. 2. Illustration of the FL framework for the given SR problem.

Algorithm 1 Federated Averaging (FedAvg)

1: **for** $t = 1, 2, \ldots$ **do**
2: **for** $i \in \mathcal{S}_t$ **do** in parallel
3: Pull $\boldsymbol{w}_t$ from central server: $\boldsymbol{w}^0_{i,t} = \boldsymbol{w}_t$
4: **for** $\tau = 1, \ldots, H$ **do**
5: Compute SGD: $\mathbf{g}^{\tau}_{i,t} = \nabla_{\boldsymbol{w}} f_n(\boldsymbol{w}^{\tau-1}_{i,t}, \zeta_{i,\tau})$
6: Update model: $\boldsymbol{w}^{\tau}_{i,t} = \boldsymbol{w}^{\tau-1}_{i,t} - \eta_t \mathbf{g}^{\tau}_{i,t}$
7: Push $\boldsymbol{w}^H_{i,t}$
8: **Federated Averaging**: $\boldsymbol{w}_{t+1} = \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} \boldsymbol{w}^H_{i,t}$

II. Problem Definition

We denote all vectors by italic bold lowercase characters and all matrices by italic bold uppercase characters.

In the scope of this work, our target is to obtain the optimal OBSS/PD threshold for SR operation that maximize the network throughput. To this end, we aim to design a neural network model which predicts the throughput of each STA in the given BSS for a chosen OBSS/PD threshold from a certain range, thus one can tune the OBSS/PD threshold by using NN architecture.

To train the NN model we employ the federated learning framework in the following way; we call a Wi-Fi deployment with specific characteristics such as node locations and number of interfering BSSs as a context. We consider n contexts in total, where for each context, we have s_i STAs per AP for context i. We also define a as the maximum number of access point and b as the maximum number of STAs per AP. Note that all contexts may have different number of STAs per AP. We also have interference sensed by APs, RSSI of the STAs assigned to $\mathrm{AP_A}$, and the average SINR of each STA in BSS_A. We can control control threshold $\tau_i \in \{-82, -81, \ldots, -62\}$. To simplify the problem as in the competition, we can only change the threshold of the BSS_A, and all other BSSs' thresholds are fixed to -82 dBm. Thus, we only consider the STAs in BSS_A.

Regarding the throughput analysis, we first want to remark that a given transmission is successful if the following two conditions are satisfied; 1) The power sensed at the receiver from the frame being decoded remains above the CCA/CS threshold, 2) Signal-to-Interference-plus-Noise Ratio (SINR) stays above the capture effect (CE) threshold [4]. Throughput of the successful transmission are determined according to modulation coding scheme (MCS) based on the SINR value and the sensed power, received signal strength indicator (RSSI). To this end, a look up table, MCS table, is used to map SINR and RSSI values to pair of modulation scheme and encoding rate.

Let $t^{(i)}_{j,\tau_i} \in \mathbb{R}$ be the throughput of j^{th} STA in the BSS_A of the i^{th} context, where threshold of the BSS_A is chosen as τ. For context i, our objective is to find τ_i that maximizes the throughput for all STAs in the BSS_A of context i, i.e,

$$\arg\max_{\boldsymbol{\tau}} \sum_{i=1}^{n} \sum_{j=1}^{s_i} t^{(i)}_{j,\tau_i}, \tag{4}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_n]^T$, s_i is the number of STAs connected to each AP (or the number of STAs in each BSS) in context i. Having the knowledge of throughputs $t^{(i)}_{j,\tau'}, \forall i, j, \tau\prime$, which is not likely, one can easily calculate $\boldsymbol{\tau}$ using (4). Thus, to determine the best threshold for each context i, we estimate $t^{(i)}_{j,\tau'}$ via $\hat{t}^{(i)}_{j,\tau_i}$ for all STA j and threshold τ' combinations.

In the federated learning setting, we model each context as a node, where their data consist of simulations with different thresholds. We assume these nodes cannot communicate with each other, but communicate with a parameter server in rounds, which aggregates the weights of the nodes to create. Then, the parameter server distributes the updated global model to the clients.

III. Methodology

In this section, we first define our multi-output objective function and our neural network model.

A. Neural Network with a Multi-Output Regression Objective

Here, we describe the neural network model and objective function, which we use in all our clients.

We cannot directly calculate or know the throughput $t^{(i)}_{j,\tau}$. Thus, we estimate it via a model $\hat{t}^{(i)}_{j,\tau} = f^{(i)}_{j,\tau}$, where i is the context index, j is the index of STA connected to $\mathrm{AP_A}$ and τ is the threshold. Moreover, estimating one STA's throughput is highly related to estimating another STA's throughput in the same context. Thus, to exploit this relation, we formulate the throughput regression problem as multi-output regression, as the following:

$$\boldsymbol{f}^{(i)}_{\tau}(\boldsymbol{x}^{(i)}_{\tau}, \boldsymbol{W}_i) = \left[f^{(i)}_{1,\tau}\ f^{(i)}_{2,\tau} \ldots f^{(i)}_{b,\tau}\right]^T, \tag{5}$$

where i is the context index, $\boldsymbol{x}^{(i)}_{\tau}$ is the input vector and $\boldsymbol{W}_i$ is the neural network weights of the model at context i.

Here, the input vector $\boldsymbol{x}^{(i)}_{\tau} \in \mathbb{R}^{4b+a}$ includes each STA's features in order (for STAs in BSS_A). Each STA's features are as the following: interference sensed by APs, RSSI, the average SINR and the threshold, respectively. When a context has less than b STA per AP. We zero pad for the remaining places until the vector reaches the maximum in the dataset. This is possible since a (the maximum number of APs) and b (the maximum number of STAs per AP) are fixed.

Since every context may have different number of STAs per AP, we mask the nonexistent STAs as the following:

$$f^{(i)}_{k,\tau} = \hat{t}^{(i)}_{k,\tau} = t^{(i)}_{k,\tau} = 0, \quad \forall k \in \{s_i + 1, \ldots, b\}.$$

This way, we do not backpropagate any loss for nonexistent STAs and become able to train our model for variable number of STAs per AP for every context.

Then, we define the ground truth vector as:

$$\boldsymbol{t}^{(i)}_{\tau} = \left[t^{(i)}_{1,\tau}\ t^{(i)}_{2,\tau}\ \ldots\ t^{(i)}_{b,\tau}\right]^T$$

For the context i (local node), our objective is to minimize mean-squared error for regression task for any $(\boldsymbol{x}^{(i)}_{\tau}, \boldsymbol{t}^{(i)}_{\tau})$ data point among all contexts, i.e.,

$$\arg\min_{\boldsymbol{W}_i} \sum_{\forall \tau, i} \left|\left| \boldsymbol{f}^{(i)}_{\tau}(\boldsymbol{x}^{(i)}_{\tau}, \boldsymbol{W}_i) - \boldsymbol{t}^{(i)}_{\tau} \right|\right|_2^2.$$

We use a feed-forward neural network with one hidden layer as our model $\boldsymbol{f}^{(i)}_{\tau}(\boldsymbol{x}^{(i)}_{\tau}, \boldsymbol{W}_i)$ with weights $\boldsymbol{W}_i = \left[\boldsymbol{W}^{(1)}_i\ \boldsymbol{W}^{(2)}_i\right]$, where $\boldsymbol{f}^{(i)}_{\tau}(\boldsymbol{x}^{(i)}_{\tau}, \boldsymbol{W}_i) = \boldsymbol{W}^{(2)}_i \mathrm{ReLU}(\boldsymbol{W}^{(1)}_i \boldsymbol{x}^{(i)}_{\tau})$, $\boldsymbol{W}^{(2)}_i \in \mathbb{R}^{b\times h}$ and $\boldsymbol{W}^{(1)}_i \in \mathbb{R}^{h\times(a+3b)}$. As seen, we use rectified linear unit (ReLU) as our activation function. Note that this neural network can easily be generalized to a neural network with multiple hidden layers, but in our case, the neural network with only 1 hidden layer have worked the best on the validation set.

In the following section, we describe how we combine these local models.

B. FL for Training

To train our NN architecture for throughput estimation we employ the FL framework particularly FedAvg illustrated in 1. To this end, we consider each AP of interest in a different context as a participant of the FL framework. Further, for each AP of interest the local dataset consist of the throughput, for each STAs of interest, and all the possible OBSS/PD thresholds[1]. FL framework for the given SR is illustrated in Fig. 2.

In general, the participation ratio, the fraction of the users participating model update in each round, and the number of local iterations are two important factors for the performance of the FL. Unless a further communication constraint is introduced we consider full participation for the FL, that is all users participate to model averaging at each communication round.

In the scope of this work, we fixed the batchsize equal to the size of the local dataset, which is 21, and fixed the number of local iteration $H = 1$.

IV. Experiments

A. The Dataset

We only use the scenario 3 dataset from the ITU-ML5G-PS-004: Federated Learning for Spatial Reuse in a Multi-BSS

[1]The size of the local dataset is equal to 21.

competition [2]. We do not use other scenarios since scenario 3 is the most complex data. The target variable is the throughput as defined in problem definition. It contains variable number of STAs per AP and different node locations. All different node locations are combined with all possible thresholds. It has between 2 and 6 APs, between 1 and 4 STAs per AP, i.e., $a \in \{2, 3, 4, 5, 6\}$ and $s_i \in \{1, 2, 3, 4\} \forall i$. Moreover, in scenario 3 dataset, different STA/AP location configurations (up to 20 different) have been generated and data points are generated for all thresholds using these STA/AP location configurations. Note that different number of .

We split the dataset into training (80% of the contexts), first validation (10% of the contexts) and second validation (10% of the contexts) sets, for which we explain the reasoning in the following section.

B. Evaluation Methodology and Implementation Details

We evaluate the prediction results of our method via the main absolute error (MAE) metric. Recall that we estimate the throughputs by multi-output regression task and each context may have different number of throughputs to be predicted. Thus, we flatten the predictions for existing STAs before calculating the MAE. We normalize the data by minmax normalization. We use to implement the standard SGD implementation of Pytorch [5] to implement federated averaging.

We have two different validation sets, where each validation set contains the data of 100 contexts. We use the first validation set for early stopping of the global model. We evaluate our global model after every 20 rounds and stop training if no improvement in validation MAE is achieved after 100 rounds. Then, we simply choose the final global model that has the lowest validation MAE throughout the training.

We use the second validation set to perform hyperparameter tuning. We tune our method by using Tree Parzen Estimator of the Optuna library [6] and choose the model with the lowest MAE on the second validation set. Finally, we compute the throughput estimations on the test set of the competition.

C. Competition Results

Our method has achieved the first place in the ITU-ML5G-PS-004: Federated Learning for Spatial Reuse in a Multi-BSS competition. Note that the results are evaluated on a test dataset, on which the throughputs (i.e. labels) were hidden from the participants.

TABLE I
ITU-ML5G-PS-004 COMPETITION RESULTS

Team	MAE (Mbps)
FedIPC (Ours)	**5.8572**
FederationS	6.5534
WirelessAI	8.9130

Table I shows the results of the ITU-ML5GPS-004: Federated Learning for Spatial Reuse in a Multi-BSS competition. Our method has achieved the minimum MAE in the test scenario. Our method has 0.6962 less MAE than the team FederationS' model and 3.0558 less MAE than the team WirelessAI's method.

Thanks to the multi-output structure, our model can learn common representations as predicting one STA's output connected to the same AP is highly similar to the predicting other STAs' outputs [7]. Moreover, since the sum of throughputs that are connected to the same AP is bounded, the multi-output structure allows to distribute the total possible throughput from an AP to its STAs.

V. CONCLUSION

We have introduced a federated learning based model for the spatial reuse problem, where each deployment is considered a node. We have employed a neural network model with multi-output objective for variable number of outputs. Our method has achieved the best performance in the test scenario of ITU-ML5GPS-004: Federated Learning for Spatial Reuse in a Multi-BSS competition.

[2]https://challenge.aiforgood.itu.int/match/matchitem/37

REFERENCES

[1] "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Enhancements for high-efficiency wlan," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, pp. 1–767, 2021.
[2] F. Wilhelmi, S. B. Muñoz, C. Cano, I. Selinis, and B. Bellalta, "Spatial reuse in ieee 802.11ax wlans," 2021.
[3] 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*, ser. Proceedings of Machine Learning Research, vol. 54. Fort Lauderdale, FL, USA: PMLR, Apr 2017, pp. 1273–1282.
[4] F. Wilhelmi, S. Barrachina-Muñoz, and B. Bellalta, "On the performance of the spatial reuse operation in ieee 802.11ax wlans," 2019.
[5] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[6] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.
[7] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.