Federated Learning for Distributed Sensing-aided Beam Prediction in 5G Networks

Adwitiya Pratap Singh
Hughes Systique Corporation
Gurgaon, Haryana, India
e-mail: Adwitiya.singh@hsc.com

Chitwan Arora
Hughes Systique Corporation
Gurgaon, Haryana, India
e-mail: chitwan.arora@hsc.com

Abheek Saha
Hughes Systique Corporation
Gurgaon, Haryana, India
e-mail: abheek.saha@hsc.com

***Abstract*— The increasing demands for higher data rates have caused newer communication systems to move towards higher frequency bands. However, during the initial network access, the user faces a problem of high beam selection, due to the rich scattering environment and the large number of possible beams. For high mobility and low latency applications, such as vehicular communications, high beam selection overhead is a very big problem. Sensing-aided beam prediction using environmental sensing information as well as telemetry data can be a possible solution to this issue. In this paper, a novel approach is suggested that combines real-time series Global Positioning System (GPS) data, as well as terrain related data for beam selection. Using the DeepSense dataset, we demonstrate that distributed machine learning algorithms, while being computationally tractable, can choose the top N beams with an accuracy that is comparable to that of centralized learning, but faster than it. The novelty of our work lies in the usage of this data set to simulate federated learning and trying different techniques to increase accuracy.**

***Keywords-Wireless Technology; Artificial Intelligence; Deep Learning; Federated learning*.**

I. INTRODUCTION

Current and future communication systems are moving to higher frequency bands. The large available bandwidth at the high frequency bands enables these systems to satisfy the increasing data rate demands of the emerging applications, such as autonomous driving, edge computing, and mixed reality [1]. These systems require the deployment of large directional antennas at both the Transmitter (Tx) and Receiver (Rx). Using directed beams to connect to the network introduces a new problem, which is choosing the optimal beam from the array of beams present at the transmitter. The overhead for the exhaustive scan to find the beam is way too high for applications that need low latency, hence we have our pain point. The way we are moving towards solving this problem is machine learning for optimization and forecasting the beams.

Sensing aided beam prediction seems to be the foot in the right direction: The mm-wave communication dependence on Line-Of-Sight (LOS) links between Tx and Rx really brings into play the sensory aid that can be provided by sensors on the transmitter and the receiver side. With the aid of GPS and image sensors, the transmitters can decide in which direction to point their beams by seeing the traffic distribution and identifying the receivers through visual sensors. This will narrow down the search done by the exhaustive scan during the initial access (as described in [2]).

Recent work on sensing-aided beam prediction has shown unprecedented results in using the sensory data, such as Red, green, blue (RGB) images, LIDAR, radar, and GPS positions for the beam prediction problem. However, the previous research is mainly done on synthetic datasets (datasets which have data that have been simulated or created virtually). While these datasets provide us insight into how the real time model would perform, there is still a disparity between modelled performance and real-time performance. Some features, such as obstruction and time of day can only be simulated on a real time dataset. This is what we are trying to achieve in this dataset. In this research paper, we will commence by reviewing the previous work conducted in this area in Section 2, followed by an in-depth examination of the problem in Section 3. Section 4 will focus on the discussion of federated learning and the distinct aggregation methods employed. Subsequently, we will present our solution implementations and results in Section 5 and Section 6, respectively. We conclude our work in Section 7.

II. RELATED WORK

There has been previous work done on synthetic dataset. In [3], the authors found out that in a raytracing implementation the deep neural network model was able to accurately predict the beam parameters up to 90%. The paper also investigated how multiple Remote Radio Heads (RRH) working together could be used to increase prediction accuracy and how they could be implemented using a hybrid edge cloud model.

The authors of [4] analysed multiple types of Deep Neural Networks (DNN). With the use of multi-modal data such as LIDAR and using separate machine learning models, they achieved an accuracy of 91.2%.

The authors of [5] propose a beam selection model based on Convoluted Neural Network (CNN). Their CNN model for the latter should contain 6 layers (2D) and 1 linear layer. GPS data was used at this point, plus the added four linear

 ISBN: 978-1-68558-068-1

layers. Their results showed 96.9% accuracy in the top 10 accuracy.

The authors in [6] focused on LIDAR data. They also explored the use of federated learning using different clients as well as using CNNs on the different nodes. The LIDAR used was from mounted sensors of the vehicles.

The authors in [7] tested the federated network used for mmWave beam-selection against a backdoor attack algorithm. Their attack basically consisted of creating obstacles on the road at specific locations. The main purpose of the attack was two-fold (1) to force the model to output a beam in a desired direction (2) to send a low signal strength beam.

In [19], the authors propose a distributed learning framework that leverages multiple vehicles as clients, each equipped with mmWave communication capabilities. The paper explores the effectiveness of this approach and demonstrates its ability to achieve accurate beam selection in vehicular mmWave systems.

Compared to the previous research, the novelty of our work is two-fold. As opposed to the works cited above, we have combined time series GPS data and stacked it with image data from the infrastructure and measure the beam selection accuracy. This will provide us with results that we can expect during practical deployment.

III. PROBLEM OVERVIEW

In an mm-wave wireless network, beamforming is an important technique used to improve the efficiency and capacity of the network. Beamforming involves adjusting the directionality of the antenna beams to focus the signal towards the intended receiver, rather than broadcasting it in all directions. This technique can be particularly effective in dense urban environments, where there are many obstacles and scattering sources.

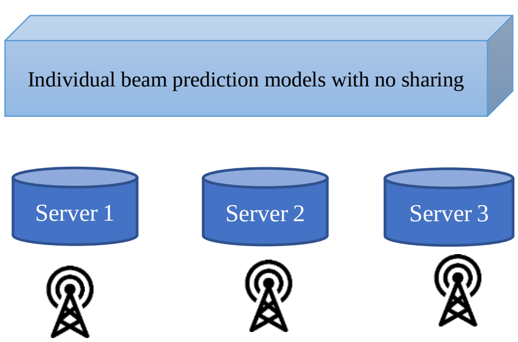

Figure 1. Centralized Beam-selection using AI.

The selection of the optimal beam for a given user is a rich target for machine learning based algorithms since there is no deterministic way to achieve this other than an exhaustive search [8]. In the first generation of machine learning algorithms, as shown in Fig. 1, gNodeB would be running in isolation. The data would be fed separately, and models would not be trained on each other datasets which would not allow the models to be apprised of different traffic distributions that are viewed by neighboring gNodeBs.

In a real-life deployment, each gNodeB would have a view of only a specific location/scenario. To create and run a centralized model, the gNodeB should have access to all possible data or scenarios. However, it would be prohibitively expensive to get the data in one centralized place. Therefore, the practical solution would be to move towards a distributed model. To properly allocate the beam index in a 5G deployment, a gNodeB can use the data of other gNodeBs around it. By sharing information with nearby gNodeBs, each can better understand the overall network conditions and adjust its beamforming accordingly. Data sharing requires nodes to collaborate even though their data may be different in terms of distribution, quality, and quantity.

However, there are significant challenges to the data sharing approach as well. Non-linear aggregation can cause the model to move in the opposite direction of the actual convergence point, so an optimized aggregation technique must be implemented. Sharing data in real-time also requires a certain amount of bandwidth that sometimes cannot be allocated due to external non-controllable factors [9]. Data sharing also brings about the risk of breaches in the network as discussed in [10].

In the context of V2I (Vehicle to Infrastructure) communication, data sharing has several advantages over centralized learning, such as the reduction volume of data and consequently latency. Furthermore, since each device can participate in the training process without the requirement for a robust central server, computing resources are employed more effectively. For instance, gNodeB can use a trained model to choose the appropriate beam for each car in an area where it detects numerous vehicles. Like this, gNodeB can utilize its learned model to modify its beamforming to avoid a certain direction if it detects high interference in that direction.

IV. FEDERATED LEARNING

A. Approach to using Federated Learning

Federated learning addresses the challenges as mentioned in the problem overview section. In our solution, as shown in Fig. 2, the gNodeB will identify the nearby towers and let them know they have similar data that can aid their beam prediction algorithms as well. There are known challenges for handling different modalities of data and optimizing the training process for every node is a big challenge [11] which we discuss in the next section.

 ISBN: 978-1-68558-068-1

B. Some downsides and their solutions

Expensive communication is a huge bottleneck in federated learning networks. Since there are millions of end devices that are usually connected in the network that at any time might be aiding the global model, the computation can be slower by many magnitudes [12]. Another problem is system reliability; if the network is comprised of many end devices such as vehicles, at any given time during a training procedure, a local device can dropout due to local system failure [13], which can lead to spurious results. Another issue is scalability, which also plays a big role, as we cannot waste much time during image feature extraction. To mitigate these issues, our task is to develop an efficient extractor that is not computationally intensive. To reduce the computation magnitude, we tried one of two methods: (1) to reduce total communication rounds of federated learning and (2) to try to reduce the data that is being transferred in the network.

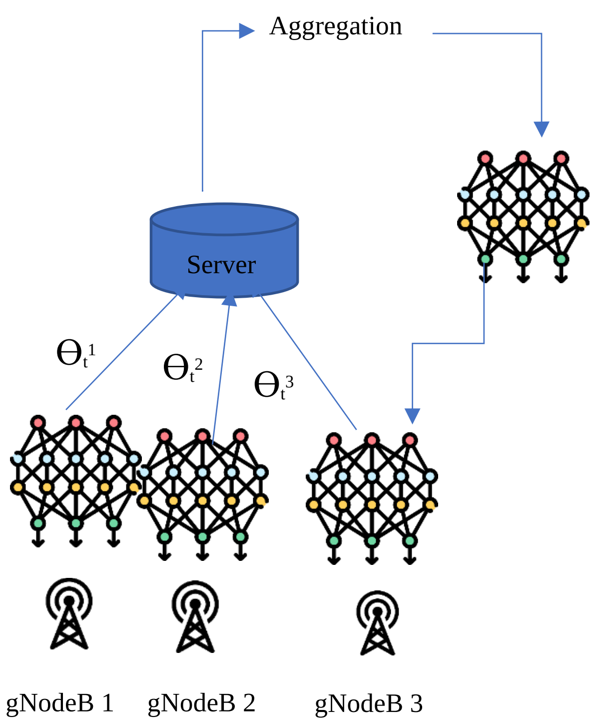

Figure 2. Federated learning from data of multiple gNodeB global model creation using aggregation techniques.

Hence, when we take the gNodeB as the end devices, we are just transferring GPS coordinates in the federated network from the vehicles to the base stations, greatly reducing the size of data transferred than if we kept vehicles as local devices. This will also increase system reliability since the chances of one gNodeB going down is significantly lower than the failure of a vehicle. To increase the scalability, we are using a computationally efficient image extractor rather than heavy transfer learning models to extract features from base station images.

C. Aggregation techniques used

While there are many different aggregations models, the best working aggregation model worked with the algorithm used by Federated Stochastic Gradient Descent (FSGD). FSGD is an alternative to averaging in which the client models are updated using Stochastic Gradient Descent (SGD) [14] before sending them to the server for aggregation. The server then combines the updated models using a weighted average. Following are the steps in this aggregation technique:

1) Local Computation

Initially each model is initialized with the same weights rather than independent initializations since according to this article [15], common initialization causes better results. The base station acts as the local server where the deep learning takes place using SGD, also the place where the cars share their GPS locations for training.

2) Model Update

In this step, each party sends its local model update to a central server. The updates are typically compressed using techniques like quantization to reduce communication overhead. The global model update is given by:

$$\Delta w(k+1) = [1]K * \sum_{i=1}^{k} (N_i|N)^* \Delta w_i(k) \qquad (1)$$

Where in equation (1) delta $w_i(k)$ is the local model update of party 'i' at iteration k, N is the total size of the data held by all parties, and K is the number of parties. The weights (N_i / N) ensure that parties with more data contribute more to the global update.

In this step, according to the figure each local server or base station needs to send the local model to the central server, this happens after all the epochs in that round of every client is completed.

3) Aggregation

In this step, the central server aggregates the global model update and sends the updated model parameters back to the parties as shown in equation (2). The aggregation can be done using different methods, such as weighted averaging, FSGD, proximal and others that have a higher privacy measure. In the case of FSGD after the weighted average is formed of the given clients then the difference between the current global model weights is computed after which we subtract the difference to move opposite to the rising gradient and towards the convergence.

$$w_{t+1} \Leftarrow \sum_{k=1}^{k} \frac{n_k}{n} w_{t+1}^k \qquad (2)$$

We contrasted FSGD against two other techniques described below.

Federated Averaging with Momentum (FedAvgM): This is an extension of the FedAvg technique that includes momentum in the aggregation step. The idea is to maintain a

 ISBN: 978-1-68558-068-1

running average of the model weights across multiple rounds to improve convergence.

Federated Proximal: Federated Proximal is a technique that uses a proximal operator to enforce sparsity in the model updates. The proximal operator is applied to the global model parameters before they are sent to the clients, and to the client updates before they are sent back to the server. This helps to reduce the communication overhead and improve the efficiency of the federated learning process.

V. IMPLEMENTATION

A. Dataset

We have implemented our model on the use case of vehicle-to-infrastructure, specifically scenario (32-34) according to the DeepSense6G dataset [16]. The testbed for getting data for these scenarios has two units: Unit 1 (a stationary unit), which acts as the base station, and Unit 2 (a vehicle), which represents the mobile user. Unit 1 is equipped with the following devices:

1) *A mmWave reciever*
2) *RGB Camera*
3) *3D LIDAR*
4) *Radar*
5) *GPS*

A scenario is a dataset collected from a combination of a transmitter (deepsense testbed 1) and receiver (vehicle) at a certain location. These scenarios differ from each other in terms of either their location or time of day. We use the different scenarios to get the independent behavior of the gNodeB.

Each scenario is a temporally ordered combination of multiple types of data, which is recorded in every 100ms. Corresponding to every timestamp there are 5 instances of image data and 2 instances of GPS data. Our algorithm will exploit the temporal information in the dataset using Gated Recurrent Units (GRUs) will be explained in the further section.

B. Model

The model receives two inputs: a sequence of position coordinates and an image. After batch normalization and Rectified linear activation unit (ReLU), the image is run through a CNN with four convolutional layers. After being flattened and passing through a fully connected layer with 128 units, batch normalization, and yet another ReLU activation function are applied to the output of the final convolutional layer.

The position coordinates are routed via a GRU layer as shown in Fig. 3 with two layers and 64 hidden units after being first embedded using a linear layer. A fully connected layer with 64 units receives the output of the GRU layer at every time step. Here we use the gated recurrent unit for processing position data since this data is temporally corelated. This step allows us to gauge the movement of the car in play. Long Short Term Memories (LSTMs) were also considered in this step but as our aim was to make this model as computationally inexpensive as possible, we went forward with the GRU, as shown in the figure below as well as the baseline solution [17].

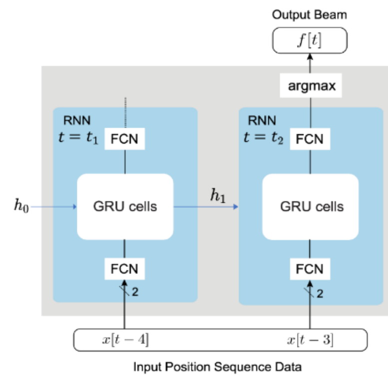

Figure 3. Visual representation of the GRU architecture used to learn the GPS data [9].

The outputs of the CNN and the GPS model are concatenated, and then passed through another fully connected layer with 128 units, followed by another ReLU activation function. Finally, the output is passed through a linear layer with number of classes units, which produces the final classification output.

The model uses dropout regularization with a rate of 0.5 to prevent overfitting, and batch normalization to speed up training and improve the model's ability to generalize to new data.

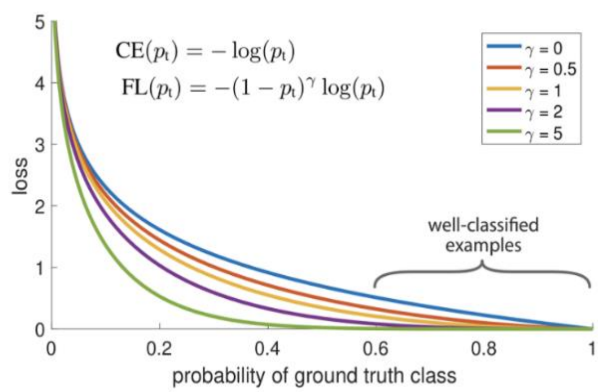

Figure 4. Focal loss representation of changing the modulating factor gamma on the loss [22].

 ISBN: 978-1-68558-068-1

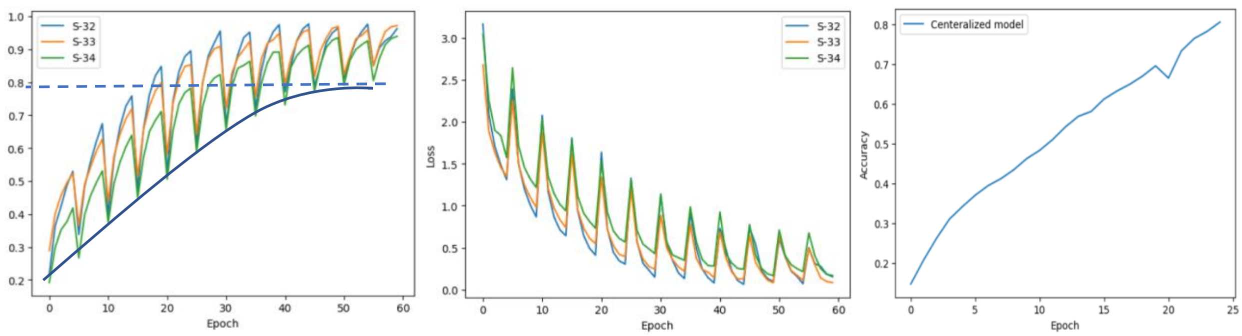

Figure 5. (a) Displays the accuracy chart of the federated learning model through all the rounds (b) shows the decreasing loss of the same federated learning model (c) the accuracy of model that has the same architecture as model before but in centralized environment.

VI. RESULTS

There are three phases in the solution that predicts the optimal beam index based on the multi modal data. In the first, data collection phase the data from local vehicles to be sent to gNodeB. Over here, the local gNodeB are initialized with the same model, without any fine tuning or aggregation from other gNodeB. In the second model update phase, every gNodeB shall send their model to the central server to be aggregated using federated stochastic gradient descent. All the models shall be aggregated to be sent to the local gNodeB for the next round. We tested out multiple local epoch numbers and came to the optimal number of 5 epoch per round. In the last phase, the updated model is sent to the local gNodeB to help implement the beam selection using the local data.

We can see in Fig. 5 that, after 10 rounds (each containing 5 epochs), the accuracy seems to be stagnating, as noted by the best fit line. We are emphasizing the minimum accuracy amongst the peaks since that is the result after aggregation. This dip in accuracy is due to the new scenario data weights that is introduced to the global model, it maxes out at 80% accuracy in beam selection. To understand why this happened we compared the baseline model to the centralized as well as centralized multi modal model to find the disparity that we will face in accuracy.

The centralized implementation is identical to the federated model except that we used the entire dataset at one node to train the model at once. Since we have a non-IID dataset this is better in terms of accuracy. But as we move towards the real-world application, the processing time consumed in training the entire dataset at once will incur a high latency. As seen in Table I, the best federated model results do lag the centralized model, but it covers in time to process, since parallel processing of three models at three different nodes allowed the model to train 37% faster on the CPU. This would be increased even further if the data is loaded on to the GPU.

A further consideration is that in every scenario there was a different amount of data available to it, and since the data was already non-Independent and identically distributed (non-IID) we used focal loss to penalize our model. In the focal loss as seen below the modulating factor reduces the contribution to the loss from easier examples such as ones which have high frequency in the dataset and extends the range in which an examples receive low loss [18]. We kept the modulating factor to 5 (shown in Fig. 4) as it provided us with the best results.

TABLE I. RESULTS OF DIFFERENT MODELS USED.

Models	***Top 5 of 64***	***Top 10 of 64***
Baseline model (GRU)	77 %	80%
Centeralized model	83%	90%
Federated Model	64%	80%

TABLE II. RESULTS OF DIFFERENT AGGREGATION TECHNIQUES.

Models	***Top 5 of 64***	***Top 10 of 64***
FSGD	64 %	80 %
Proximal	60 %	75 %
Fed Avg	65 %	76 %

In Table II, we can compare the different aggregation techniques used during federated learning. As mentioned before in the implementation section, we know that the federated stochastic gradient descent worked best amongst all. This can be corroborated with theory as well since FSGD is slightly immune to the non-IID imbalanced dataset since it allows for more local model updates. The use of sampling only a subset of the local data to perform the local updates helps FSGD pay less attention to outlier data, as well as making the gradient correct. This is very important when not using such a large dataset such as ours, as well as having a small number of nodes. Although the performance could be further improved if we were able to introduce more types of scenarios of V2I from the Deepsense dataset hence increasing our number of nodes.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated the use of FSGD based federated learning optimal beam selection. We used aid from sensors to allow a multi-modal model accurately predict the beams. The use of other sensors can also prove to be a viable option in sensor-aided beam prediction such as accelerometers and gyroscopes. Different aggregations techniques can be explored to analyze the resultant effect in the performance of federated learning. The federated model falls prey to overfitting if given a small number of clients, we can investigate the behavior by varying the number of active clients in federated learning.

In conclusion, sensing-aided beam prediction is a promising solution for the challenges faced by mmWave communication systems. The utilization of sensory data collected by various sensors can guide the beam management process and significantly reduce beam training overhead. In real-life deployment it is impractical to get all the data at one centralized place for training as a result federated learning can be used as a preferable training solution. Although it is noticed that the accuracy of the federated model is lesser than that of centralized model, we can see that we have a trade of between accuracy and practical realization of latency. Our work received a top 10 running accuracy score of 80%. Federated stochastic gradient descent produced the best results in terms of aggregation techniques.

REFERENCES

[1] ITU AI for Good Global Summit ", AI for Good Global Summit 2021", ITU AI For Good Challenge, https://challenge.aiforgood.itu.int/match/matchitem/72

[2] H. Lindgren and M. Hasselqvist, "The Effects of Battery Aging on the Energy Management of a Parallel Hybrid Electric Vehicle", Proceedings of the 2017 19th European Conference on Power Electronics and Applications (EPE'17 ECCE Europe), Chalmers University of Technology, 2017, pp. 1-6.

[3] C. Arora and A. Saha, "AI Based Beam Management for 5G (mmWave) at Wireless Edge", IARIA International Journal on Communications (IJCN), vol. 11, no. 3 & 4, 2021, pp. 44-53. DOI: 10.1016/j.ijcon.2021.100038

[4] J. Smith et al., "A Novel Approach to Federated Learning", IEEE Signal Processing for Wireless Communications (SPAWC), IEEE, 2019, pp. 1-6. DOI: https://doi.org/10.1109/SPAWC.2019.8815569Sdad

[5] J. Yang et al., "Federated Learning for Intelligent Vehicular Networks: Challenges and Solutions", IEEE Transactions on Vehicular Technology, vol. 71, no. 3, 2022, pp. 3079-3091. DOI: https://doi.org/10.1109/TVT.2022.3142513

[6] S. Ashish et al., "Secure Federated Learning for Next-Generation Wireless Communication Networks", IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), 2021, pp. 1-6

[7] M. Akbari and E. Mohammadi, "Risk and Advantages of Federated Learning for Health Care Data Collaboration", ResearchGate, 2020.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks", arXiv preprint arXiv:1502.02367, 2015.

[9] Y. Chen, J. Liu, M. Xu, and K. Ren, "Federated learning in mobile edge networks: A comprehensive survey", IEEE Communications Surveys & Tutorials, vol. 20, no. 3, 2018, pp. 2022-2063. DOI: 10.1109/COMST.2018.2834990

[10] M. Karimzadeh et al., "Secure Federated Learning with Homomorphic Encryption: A Review", IEEE Access, vol. 9, 2021, pp. 197789-197814. DOI: https://ieeexplore.ieee.org/document/9862973

[11] "Deep Learning-Based mmWave Beam Selection for 5G NR/6G With Sub-6 GHz Channel Information: Algorithms and Prototype Validation", ResearchGate, 2020, DOI: 10.13140/RG.2.2.15928.44809

[12] C. Van Berkel, "Multi-core for mobile phones", In Conference on Design, Automation and Test in Europe, 2009, pp. 1-6.

[13] A. Kumar Sahu, T. Li, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions", Carnegie Mellon University, Bosch Center for Artificial Intelligence, and Determined AI, 2019.

[14] J. Konečnỳ et al., "Federated learning: Strategies for improving communication efficiency", arXiv preprint arXiv:1610.05492, 2016.

[15] "CS294-163: Federated Learning", University of California, Berkeley, Electrical Engineering and Computer Science, https://inst.eecs.berkeley.edu/~cs294-163/fa19/slides/federated-learning.pdf.

[16] G. Charan Alkhateeb et al., "DeepSense 6G: large-scale real-world multi-modal sensing and communication datasets", to be available on arXiv, 2022.

[17] DeepSense6G, "Multi-Modal-Beam-Prediction-Challenge-2022-Baseline", [GitHub repository], https://github.com/DeepSense6G/Multi-Modal-Beam-Prediction-Challenge-2022-Baseline, 2022.

[18] T. Y. Lin et al., "Focal loss for dense object detection", In Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980-2988.

[19] M. Brown, C. Thompson, and B. Anderson, "A federated learning approach for beam selection in vehicular mmWave systems", In Proceedings of the IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1-6.