Federated Learning based Base Station Selection using LiDAR Data

Mohit Mehta[1], Bhagyashree Gour[1], Thushan Sivalingam[2], *Graduate Student Member, IEEE,* Vimal Bhatia[1], *Senior Member, IEEE* and Nandana Rajatheva[2], *Senior Member, IEEE*

***Abstract*—Base station (BS) selection plays an important role in establishing communication links in the millimeter-wave (mm-Wave) based communication systems. For selecting the best BS, each BS performs a handshake with the user equipment (UE). This selection is further challenging due to the increased communication overhead imposed by the handshake between multiple BSs and UEs. This paper investigates BS selection for autonomous vehicles (AVs) using deep learning (DL). Due to the rigid foundation of signal processing algorithms in statistics and information theory, they do not account for non-linearities and imperfections in the system, which DL-based communication systems can mitigate. Further, federated learning (FL) is applied where BS broadcasts its position to all nodes to reduce communication overhead. The paper also describes the dataset generation using the ray-tracing technique and detection and ranging (LiDAR) sensor. Finally, the simulation results verify that the proposed algorithm performs considerably better, with an accuracy of 1.7 times better than global positioning system (GPS) based selection and a reduction of 96.38% in overall data size using FL-based selection, thereby reducing communication overhead significantly. The generality of the proposed model has been further tested by using techniques like Transfer Learning(TL) for variation in the number of samples for training, the city used for simulation, and the number of BSs.**

***Index Terms*—Autonomous vehicles, deep learning, federated learning, LiDAR, mm-Wave.**

I. Introduction

Rapid development in autonomous vehicles (AVs) and their communication with infrastructure, known as vehicle-to-infrastructure (V2I) communication, has increased safety and also reduced fuel consumption [1]. The driverless vehicle eliminates driver error, significantly reducing fatal accidents by 90% [1]. As traffic congestion decreases and lane capacity increases, there is a reduction in the travel time of AV. The sensor data collected from AV can be shared using millimeter-wave (mm-Wave) communication in 5G [2]. The BS selection is made for the movement of the AV through the city. A Multihop cellular network is conventionally used as the strategy for base station (BS) selection [3]. Leveraging the side information like global positioning system (GPS) coordinates and light detection and ranging (LiDAR) data could aid in better estimation, thereby reducing communication overhead. [4].

Machine learning (ML) is one of the most promising technologies that have been proven conducive to solving numerous telecommunications problems, like physical layer optimizations, network management, and conceives to support smart radio terminals [5]. Further, Klautau et al. [6], [7] suggest that ML helps in learning the diverse characteristics of both users and the environment, thus supporting the establishment of communication links for AV. The authors in [6] investigate beam selection for V2I with a realistic 5G mm-Wave channel model combining vehicle traffic and a ray-tracing simulator with mobility. Likewise, [7] presents a line-of-sight detection-based beam selection for mm-Wave vehicular communication. The above two works exemplify the significance of realistic channel data for ML-based models. The low accuracy received after using conventional methods such as shortest distance further motivates us to develop deep learning (DL) based model for selecting BS. The convolutional neural network (CNN) model is preferred where the wireless environment is complex and evolving due to CNN's ability to learn diverse data with good accuracy. With the evolution of mobile edge computing, federated learning (FL) is an emerging technology that helps in fortifying user privacy and takes advantage of user participation, where the training takes place across multiple decentralized edge devices (vehicles) [8]. They learn a shared model while preserving the training data simultaneously. Thus in this way, the data is kept private, and the communication overhead is reduced [4].

A federated mm-Wave beam selection using LiDAR data is presented in [9], where connected vehicles with a single BS train a deep neural network model in a collaborative manner using a local LiDAR sensor. In contrast, we present a novel method for selecting the best BS from multiple BSs within 100 meters range from a vehicle while keeping track of traffic conditions. A CNN architecture is proposed along with developing the dataset (LiDAR and GPS) and the dataset pre-processing technique for the data-driven BS selection. Further, implementing FL on multiple BSs is considered instead of single BSs as done in [9], which is more practical in 5G deployment. FL shows outstanding results in reducing overall data size as the data transferred from vehicle to BSs is considerably reduced. Furthermore, FL is implemented on the same dataset to analyze LiDAR-based model robustness.

The paper is organized as follows: Section II introduces the system model. Section III describes the data generation methodology. Section IV illustrates our proposed CNN model for mm-Wave BS selection. Simulation results are reported in Section V. Finally, Section VI concludes the paper and describes the future work.

Notation: Scalars, matrices, and sets are represented by letters (a, A), lower case bold face (**a**), and upper case bold face (**A**) letters respectively.

[1]Discipline of Electrical Engineering at IIT Indore, Indore, Madhya Pradesh, 453552, India. (**email:** mohitmehta013@gmail.com, bhagyashrigour56351@gmail.com and vbhatia@iiti.ac.in).

[2]Department of Information Technology and Electrical Engineering at University of Oulu, Oulu, 90570, Finland. (**email:** thushan.sivalingam@oulu.fi, nandana.rajatheva@oulu.fi).

II. System Model

Ray tracing is a promising simulation strategy for 5G mm-Wave multiple input multiple output (MIMO) channels [6], which provides accurate results, however the computational cost increases exponentially with the number of reflections and diffractions. For good ray tracing accuracy, the scenario should have detailed specifications (geometry, material, size) of buildings and vehicles, which makes it a site-specific simulation [6]. We consider a simple yet effective and scalable system model for simulating real-time traffic and analyzing the communication system. An open-source robotics simulator, Webots [10], and simulation for urban mobility (SUMO) is used as the traffic simulator, coupled with Matlab to assess communication characteristics using accurate ray tracing [11]. As shown in Fig. 1, the system model consists of one vehicle and three BSs, which are within 100 meters range of the target vehicle in model of a downtown Rossyln, Virginia, as it is heavily urbanized. Friis equation is used to find the ideal power received P_{rx} at an antenna from basic information about the transmission and is given as

$$P_{rx}(dB) = P_{tx} + G_{tx} + G_{rx} + TPL \tag{1}$$

where G_{tx} and G_{rx} are transmit antenna gain and receive antenna gain respectively in dB. P_{tx} is the power gain of transmitting antenna in dB and TPL is the total power loss in dB.

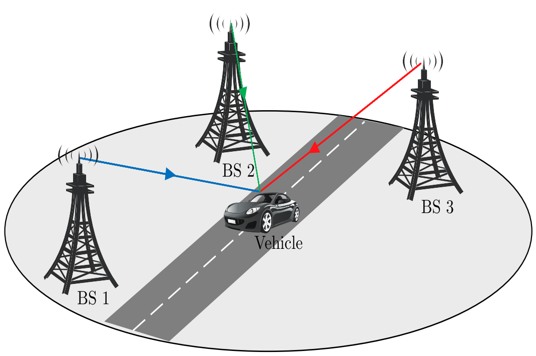

Fig. 1: System Model

A. Propagation model

The propagation model facilitates the prediction of propagation loss and attenuation occurring in the signal traveling through the environment. The ray-tracing model used in simulation computes multiple propagation paths [12]. The model learns the line of sight (LoS) path by launching ray from the transmitter (Tx) to the receiver (Rx). The shooting and bouncing rays (SBR) method is used for non-line of sight (NLoS) transmission as the computational complexity increases linearly with the number of reflections. The model calculates losses using a Fresnel equation for each reflection. The power losses include free-space path losses (FSPL) and reflection losses (RLs).

$$TPL = FSPL + RL \tag{2}$$

As the ray interacts with the surface at some angle, RL is calculated using Fresnel's equation. The ray-tracing model computes reflection loss by using the reflection matrix computations. For the current simulation, the materials are considered as "perfect reflector"; hence reflection loss is equal to zero. The FSPL in the far-field of the Tx (in decibels) as given below:

$$FSPL = 20\log_{10}((4\pi r)/\lambda) \tag{3}$$

where r is the distance traveled by the ray from Tx to Rx antenna, and λ is the wavelength.

III. Data Generation

As an example implementation, we consider the downtown Rossyln, Virginia, for generating the dataset and pre-processing methodology as shown in Fig. 2 [6]. The city-wide map has been imported from OpenStreetMaps [13] in the form of OSM files, and the 3D world is constructed using Webots internal importer. After creating the 3D map, the traffic is generated using SUMO.

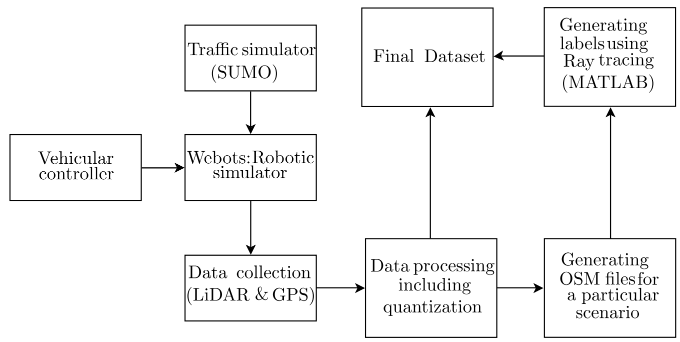

Fig. 2: Data generation process

We consider 60 vehicles in the 3D model, with each vehicle consisting of a LiDAR sensor - Velodyne 64E and three GPS sensors. The Velodyne HDL 64E is a 64-layer LiDAR with a range of up to 120 meters and a field of view of 360 degrees which returns 4500 points per layer per scan. The model of the Velodyne HDL 64E contains a Gaussian noise with a standard deviation of 0.02 meters and a rotating head [10]. We also consider that the GPS is mounted on the vehicle and is devoid of any noise or errors [10]. The LiDAR sensor is mounted on the vehicle's roof. The GPS is mounted at the front, center, and rear to efficiently retrieve the vehicle's position and orientation with vehicle bounding objects taken as rectangles.

A. MATLAB: Ray tracing

Ray tracing is used to gauge the propagation path and the losses accurately. The transmitter taken is 4×4 uniform rectangular array (URA) with element spacing of 0.1 meters in both x and y directions. As shown in Fig. 3, the Tx is located at an altitude of 5 meters surface of the building or terrain, radiating at a frequency of 60GHz at 1W [6]. Ray tracing with the SBR method is used as the propagation model. In medium angular separation, rays have an angular separation in the range [0.4956, 0.5923] measured in degrees so that the model launches 163,842 rays. The maximum number of reflections considered is 2, with both building material (buildings and vehicles) and terrain material as a perfect reflector. The same can be generalized to different cities.

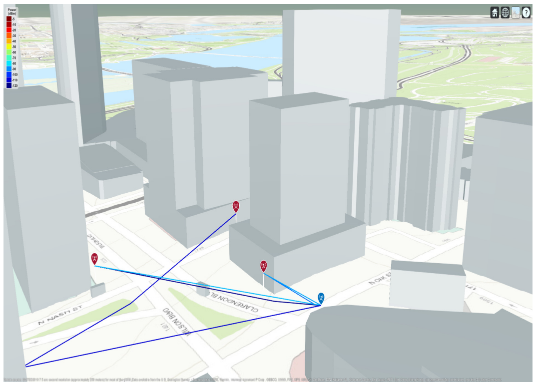

Fig. 3: Ray Tracing in MATLAB

IV. Deep Learning for BS Selection in V2I

This section discusses the pre-processing of the generated dataset, followed by the shortest distance method performance for selecting the best BS. Also, we will discuss the DL model using different sensor data to intelligently select the best BS, which will be further used to combine DL with FL for LiDAR data to reduce the communication overhead.

A. Pre-processing the data

The captured LiDAR dataset is quantized following the parameter described in [6] with quantization steps of 1.0 meters in the x-plane, 0.5 meters in the y-plane, and 1.0 meters in the z-plane. Resultant input shape of [10, 240, 240] according to [y, x, z] with the x and z restricted by scanning range of Velodyne 64E LiDAR sensor, i.e., 120 meters and y range is taken to be 5 meters. Total samples are divided into training, validation, and testing in the ratio of 8:1:1.

B. Based on shortest distance

For benchmarking the dataset, the BS is selected based on the shortest distance of the vehicle from all possible BSs in the 100m range using coordinates of BS and vehicle. The overall accuracy of 27.35% for BS selection is achieved by using this method, however, the predictability of best BS decreases for NLoS channels compared to LoS channels. This low accuracy using the shortest distance motivates us to apply DL on the selection of BS as the environment is complex.

C. Deep learning on GPS data

Since installing a GPS is easy and is now coming as a standard feature in vehicles, a compact deep neural network (DNN) model on GPS data with overall 339 parameters and three hidden layers of size 8, 16, 8 is considered. The activation function used here is the parametric rectified linear unit (ReLU) with a negative slope set to 0.25, as given below:

$$\begin{aligned} f(y_i) &= y_i, \quad \text{if } y_i \geq 0 \\ f(y_i) &= a_i y_i, \quad \text{if } y_i \leq 0 \end{aligned} \tag{4}$$

For efficient training of the model, cross-entropy loss has been used, shown in (5), coupled with Adam Optimizer. As the system model considers both LoS and NLoS channels, the model's accuracy depends on the blockage probability, which is heavily influenced by traffic statistics, large vehicles, and antenna height.

$$L_{CE} = -\sum_{i=1}^{n} t_i \log_{10}(p_i) \tag{5}$$

where n is the total number of classes, here 3 or 5 is used; t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

An accuracy of 35.54% is achieved for selecting the best BS from multiple BSs in 100 meters range using GPS data in DNN model, which is higher than shortest distance.

D. Deep Learning on LiDAR data

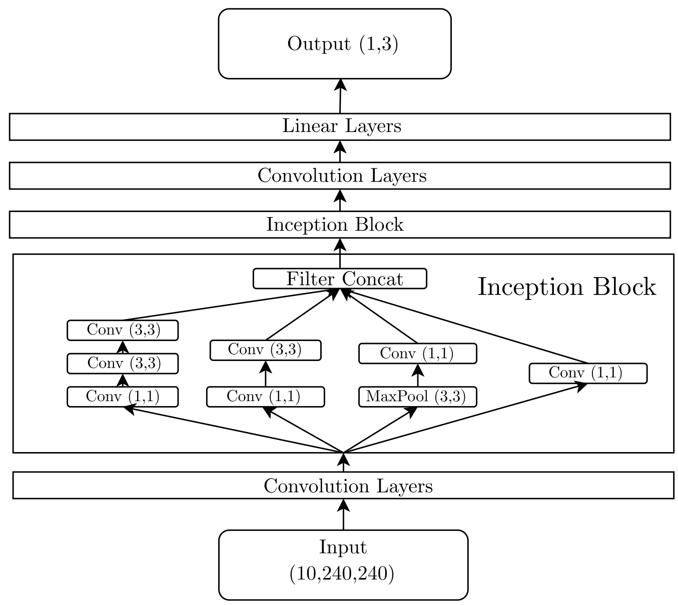

Fig. 4: The proposed LiDAR CNN model with inception block

Since LiDAR gives us more accurate realistic 3D data in respect to surroundings as compared to GPS, we applied the proposed CNN architecture on LiDAR data, as given in Fig. 4. The initial convolution layers feature a high kernel (two (13, 13) and two (7, 7)) sizes to reduce sparsity in the LiDAR data while also reducing the vector size at the same time. Later, Google's Inception-inspired model architecture is used to not only expand the network in depth but also in width [14]. It provides a novel architecture to reduce the computational cost for the same accuracy. The model contains two inception blocks whose output is passed to the filter concatenation layer, which concatenates all the output in the filter dimension. This renders the output channel four times than that of the convolution output channel. Finally, there are other convolution layers (two (7, 7) and two (3, 3) kernels) to reduce again the dimension of the model followed by a linear layer to convert the vector to the required dimensions. The output of the linear layer is passed through a softmax layer, where the output softmax layer computes the probabilities for all the BS based on the training data. The size of the output vector depends on the number of surrounding BSs, in our case $(1, 3)$ and $(1, 5)$. Also, it can be modeled based on the number of BSs. Finally, our model chooses the highest probability as the output best BS. We compute the accuracy over the testing dataset, where the predicted BS is compared with the true values (actual label) generated by the ray tracing tool (MATLAB). In order to efficiently train the model, cross-entropy loss has been used, shown in (5) with optimizer being Adam, tuned with weight decay of 1×10^{-4} and learning rate of 3.63×10^{-4}. The proposed model achieved state-of-the-art accuracy of 60.55% with 20 epochs and automatic mixed-precision set to 16 floating bits.

E. Federated Learning

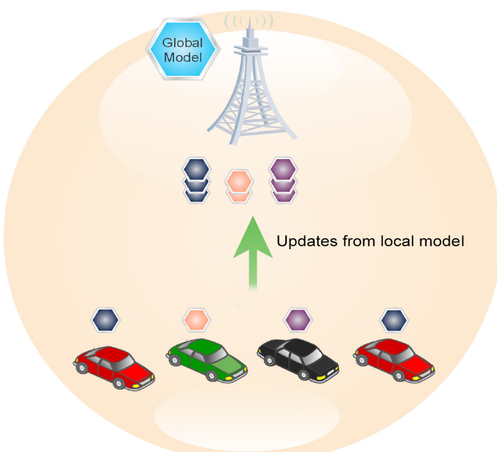

Fig. 5: Federated Learning

FL aims to predict a realistic model which accounts for the local data without sharing it with the server. FL helps the CNN model gain experience from a vast range of data located at different sites. The vehicles use federated averaging (FedAvg), where a global model is sent to the vehicles from the BS for each round, and the vehicles perform batch gradient descent updates based on their local datasets. Let $\boldsymbol{\theta}$ be the weights of the model used in training and $\boldsymbol{V}$ be the overall vehicles present, with t being overall number of vehicles. Therefore, $\boldsymbol{\theta}_{\boldsymbol{v}}^{\boldsymbol{i}}$ represents weights of model allocated to vehicle v at i communication round, where each communication round represents an aggregation of weights of different vehicles using mean at base stations and synchronizing them. Algorithm 1 represents the training loop for federated learning for n communication rounds between the base station and vehicles.

The local updates of the trained model of each device are sent to the global server, where it is aggregated with other device updates to improve the global model, as shown in Fig. 5. The updated global model is then used to train the local devices for the next round. The following results are computed using the mean aggregation method for aggregating

Algorithm 1: FedAvg for LiDAR-assisted BS selection

Init: *Initial parameters* $\boldsymbol{\theta}_{\boldsymbol{v}}^{(\mathbf{0})} = \boldsymbol{\theta}^{(\mathbf{0})}, \forall v \in \boldsymbol{V}$;
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** t **do**
    Each vehicle perform k local epochs using batch gradient descent ;
  Each vehicle v sends $\boldsymbol{\theta}_{\boldsymbol{v}}^{(\boldsymbol{i})}$ to the base station;
  BS computes $\boldsymbol{\theta}^{(\boldsymbol{i})} = \frac{\sum_{v=1}^{t} \boldsymbol{\theta}_{\boldsymbol{v}}^{\boldsymbol{i}}}{t}$;
  BS distributes $\boldsymbol{\theta}^{(\boldsymbol{i})}$ such that $\boldsymbol{\theta}_{\boldsymbol{v}}^{(\boldsymbol{i})} = \boldsymbol{\theta}^{(\boldsymbol{i})}, \forall v \in \boldsymbol{V}$

the updates [9]. To train the CNN classifier, the cross-entropy loss is calculated using (5) with an Adam optimizer with an initial learning rate of 10^3 and a batch size of 64. The models are trained for two epochs per data set, with overall communication rounds being 10.

V. Results

The realistic dataset is simulated using ray tracing and LiDAR sensor data to suggest the best BS by using CNN. The distribution of surrounding vehicles follows Gaussian distribution over the whole dataset (training, validation, and testing) as shown in Fig. 6. The CNN model achieves an accuracy of 60.55% using LiDAR data. Further, we have utilized FL to increase user privacy while reducing the communication overhead between users and BS. In the absence of FL, raw LiDAR data was sent to the BS for pre-processing, having 4.3958 MB in size. While using FL, only model weights are transferred to the vehicles, which are 0.1591 MB in size, thereby resulting in a 96.38% reduction in the overall data size. This increases the efficiency of the overall system model and reduces the communication overhead, keeping the accuracy almost intact, i.e., 58.51% compared to 60.55% of centralized architecture as depicted in TABLE I.

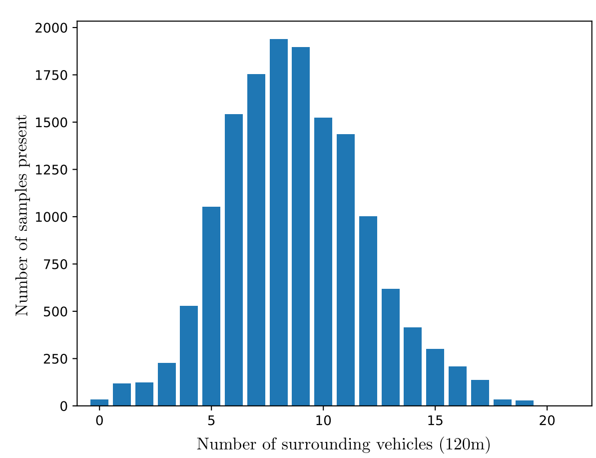

Fig. 6: Distribution of surrounding vehicles over the collected data

The relationship between the number of samples and accuracy is considered next. The number of samples for training increases from 12,000 to 24,000 and 36,000, respectively.

TABLE I: Comparison between different approaches.

Model name	Accuracy	No. of parameters
LiDAR	60.55%	28×10^3
FL on LiDAR	58.51%	28×10^3

TABLE II: Variation of accuracy with change in number of samples.

Samples	Accuracy
12,000	60.55%
24,000	64.45%
36,000	66.25%

Table II highlights the accuracy achieved on each dataset for the same epochs. The accuracy increases by increasing the number of training samples. Furthermore, to analyze and test the robustness of the model, the following experiments have been conducted.

To verify the applicability and generality of the model, we analyzed the model for 5 BSs using Transfer Learning (TL), where the number of BSs is increased from 3 to 5 and simulated around 2000 samples in which 1000 samples are used for training, and the remaining 1000 samples are used for testing. The output of the linear layer is changed from a vector size of 3 to 5 in the basic model. The main focus of the experiment is geared towards the use of TL to train the pre-trained model with 3 BSs to adapt to an increase in the number of BSs to 5. Various methodologies were used for TL, like fine-tuning the entire model or freezing the initial convolution layers and then fine-tuning the later dense layer. Table III summarizes all the different approaches and their respective accuracies.

TABLE III: Comparison between different approaches for increase in BSs from 3 to 5.

Training Methodology	Accuracy
Freezing Convolution Layers and Fine-tuning	47.60%
Fine-tuning the entire model	46.77%

Since the model is sensitive to the location, to test the robustness of the model, an experiment is performed in which the location is changed from Rossyln, Virginia, to downtown Chicago city. Around 2000 samples are simulated in which 1000 samples are used for training, and another 1000 samples are used for testing, same as in the previous experiment. Table IV reports the accuracy achieved by using both transfer learning on a pre-trained model and training from scratch using the existing model.

TABLE IV: Comparison between different approaches for change in location to downtown Chicago.

Training Methodology	Accuracy
Freezing Convolution Layers and Fine-tuning	48.33%
Training from scratch	46.67%
Fine-tuning the entire model	45.94%
Testing with pre-trained weights	37.92%

VI. Conclusion and Future work

A methodology to realistically and accurately simulate the data for BS selection is proposed in the paper. A scheme for BS selection is proposed that leverages LiDAR data in CNN to reduce the BS search overhead and achieve greater accuracy. Introducing FL to the CNN model further reduces the communication overhead as the data transferred to the BS is reduced significantly with a slight loss of accuracy. Furthermore, the given model has been tested in multiple scenarios with variations in parameters like the number of BSs, change in location, and the training data size. The proposed model can also be applied to other scenarios with the help of transfer learning. The existing model will be strengthened by increasing the complexity of the communication environment and introducing techniques such as beam selection and beam-forming as part of the future work. Also, the system will be updated to handle the handover of the vehicle signal from one BS to another BS.

References

[1] T. Litman, *Autonomous Vehicle Implementation Predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[2] N. González-Prelcic, A. Ali, V. Va, and R. W. Heath, "Millimeter-wave communication with out-of-band information," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 140–146, 2017.

[3] N. Marathe, U. Desai, and S. Merchant, "Base station selection strategy in multihop cellular networks: A new approach," in *2008 International Conference on Signal Processing, Communications and Networking*, 2008, pp. 401–404.

[4] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019, pp. 1387–1395.

[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2016.

[6] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*, 2018, pp. 1–9.

[7] A. Klautau, N. González-Prelcic, and R. W. Heath, "LiDAR data for deep learning-based mmWave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[8] W. House, "Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy," *Journal of Privacy and Confidentiality*, vol. 4, no. 2, Mar. 2013. [Online]. Available: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/623

[9] M. B. Mashhadi, M. Jankowski, T.-Y. Tung, S. Kobus, and D. Gunduz, "Federated mmWave beam selection utilizing LiDAR data," *arXiv preprint arXiv:2102.02802*, 2021.

[10] Webots, "http://www.cyberbotics.com," open-source Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[11] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using SUMO," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[12] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[13] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org , 2017.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.