Federated Learning based Base Station Selection using LiDAR data

M.Tech. Thesis

By

SHIVANI YADAV

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

Federated Learning based Base Station Selection using LiDAR data

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Technology

by

SHIVANI YADAV

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Federated Learning Based Base Station Selection using LiDAR data** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DEPARTMENT OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August 2021 to June 2023 under the supervision of Prof. Vimal Bhatia.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Shivani Y.
22-06-2023

Signature of the student with date
(SHIVANI YADAV)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

22.06.2023
Signature of the Supervisor of
M.Tech. thesis #1 (with date)
(PROF. Vimal Bhatia)

SHIVANI YADAV has successfully given his/her M.Tech. Oral Examination held on **May 16th 2023.**

22.06.2023
Signature(s) of Supervisor(s) of M.Tech. thesis
Date:

Convener, DPGC
Date: 26/06/2023

26/06/2023
Signature of PSPC Member #1
Date:

Signature of PSPC Member #2
Date: 26/06/2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who have supported and contributed to the completion of this Master's thesis. Without their assistance, guidance, and encouragement, this research would not have been possible. First and foremost, I am deeply grateful to my supervisor, **Prof. Vimal Bhatia**, for his continuous support throughout the entire research process. I am indebted to the research scholar, **Mr. Nikhil Kaler** and **b.tech student Sakshi Verma**, for being an instrumental for my work. I appreciate them for their invaluable and thorough guidance during the course of my thesis. The unwavering commitment to excellence, patience, and insightful feedback played a significant role in shaping this thesis. I would also like to extend my appreciation to the faculty members and academic staff at **Indian Institute of Technology, Indore**, particularly the **Department of Electrical Engineering**, for offering valuable resources that enhanced the quality of my study. Their dedication to fostering an atmosphere of academic rigor and intellectual growth has been instrumental in my personal and professional development. Furthermore, I am grateful to my lab mates, friends and colleagues who have provided support and encouragement throughout this demanding academic journey. Their unwavering belief in my abilities and their willingness to lend an ear or provide advice have been a constant source of inspiration. Lastly, I would like to express my deepest appreciation to my family. Their unconditional love, understanding, and encouragement have been my pillars of strength throughout my academic pursuits. Their sacrifices and belief in my potential have been the driving force behind my achievements.

In conclusion, I am humbled and honored to have received support from all these individuals and institutions. Their contributions have been invaluable, and I am truly grateful and indebted for their involvement in this research endeavor.

Abstract

The optimum selection of base station (BS) around the mobile vehicle is important for establishing communication links in the mm-Wave-based communication systems to get a reliable and low latency link between them.

For selecting the best BS, each BS performs a handshake with the mobile vehicle, using the ray tracing method power loss is calculated between them and the best BS is selected.

In this research, we have investigated the best base station out of the 3 BSs we have considered in our system model using Ray tracing method and RSSI values and further compared it with that calculated using machine learning (ML) model.

We have also applied federated learning (FL) algorithms to our ML model to reduce the communication overhead and to preserve the privacy of the user. Different FL algorithms are compared based on various parameters in our model to get the test accuracy of these algorithms, where the simulation results shows the accuracy achieved using these algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

5G	5th Generation
AV	Autonomous Vehicle
BS	Base Station
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
FedAvg	Federated Averaging
FedSGD	Federated Stochastic Gradient Descent
FL	Federated Learning
GPS	Global Positioning System
LIDAR	Light Detection And Ranging
LoS	Line-of-Sight
MIMO	Multiple Input Multiple Output
ML	Machine Learning
mm-Wave	Millimetre-wave
NLoS	Non-Line-of-Sight
OSM	Open Street Maps
RT	Ray Tracing
Rx	Receiver

SBR	Shooting and Bouncing Rays
SUMO	Simulation for Urban Mobility
Tx	Transmitter
UE	User Equipment
URA	Uniform Rectangular Array
URLLC	Ultra reliable low latency communication
V2I	Vehicle-to-Infrastructure

CHAPTER 1

Introduction

There has been a rapid development in the sector of autonomous vehicles (AV) and their communication with the environment. Autonomous vehicles eliminate the need for human intervention by using sensors to learn about the environment around them and to attentively navigate around. Driver-less cars eliminate driver error and can reduce fatal accidents to up to 90%.

AVs collect large quantities of light detection and ranging (LiDAR) data and global positioning system (GPS) data while interacting with the environment for monitoring the movement of AV and ensuring safety. This sensor data collected from AV can be shared using mm-Wave communications using the pre-eminent technology of 5G, i.e., ultra reliable and low latency communication (URLLC).

Machine learning (ML) is one of the widely used technology in this research for learning diverse characteristics of both the vehicle and the environment for establishing communication links and choosing the optimum base station (BS) for this mobile device. 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.

Federated learning (FL) is one of the emerging technologies which helps in preserving user privacy and meanwhile keeping the communication overhead to its minimum. Here the training takes place across multiple decentralized edge devices (vehicles) rather than on the central server. So, they learn the shared model while preserving the training data simultaneously. Thus in this way, the data is kept private, and the communication overhead is reduced.

In this project, we present a novel method for selecting the best BS from multiple BSs around the vehicle while keeping track of traffic conditions. A CNN architecture is proposed along with the development of the dataset (LiDAR and GPS). Further, two FL algorithms are applied to this model and different parameters of the model are compared for analysis.

1.1 Problem Statement

The problem statement revolves around selecting the best BS as determined by the power of received beams among all possible choices. The best BS is to be selected from multiple BSs by leveraging the sensor data and keeping track of traffic. The proposed model is trained to leverage LiDAR and positional data for best beam selection. For realistic calculation of communication parameters, simulation is performed using ray tracing, LIDAR, and GPS. We use an open-source robotic simulator - Webots and SUMO as our traffic simulator, followed by pre-processing and coupled with MATLAB for the generation dataset. Also, received signal strength indicator (RSSI) values have been calculated for comparison with the raytracing method.

Machine learning is being used in vehicular networks. ML is used for learning diverse characteristics of both the vehicles and the environment at different positioning to learn the model better and thus supporting the establishment of communication links for AV. Majority of ML models use centralized learning, but there is a drawback of using ML, which is that it increases communication overheads, and issue of privacy of the user stays.

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). 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.

Summarizing, in our model a novel method is approached 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 the development of 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, which is more practical in 5G deployment. Two FL approaches are applied and compared, which 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.

1.2 Thesis Outline

Chapter 1 has given a basic introduction to the need for AV, development in the sector of V2I, and the objective of the work in brief.

The remaining contents are organized as follows:

Chapter 2: This chapter contains a review of past work done in the domain of AV using LIDAR data, and RT and it widely describes the problem statement.

Chapter 3: This chapter provides background and details about the fundamentals used further in the thesis. Section 3.1 discusses the system model used, section 3.2 describes the propagation model, section 3.3 covers all the fundamentals of DL and section 3.4 introduces Federated Learning to this work.

Chapter 4: This chapter covers a description of the generation of the environment and the generation of the dataset where RT and LiDAR dataset generation is explained in detail.

Chapter 5: This chapter provides details about the different proposed approaches.

Chapter 6: This chapter covers experimental results and discussions of the proposed results.

Chapter 7: In this chapter, conclusions are made, and a discussion on the possibility of future work is presented.

CHAPTER 2

Review of Past Work and Problem Formulation

2.1 Literature Survey

To assess the overall dynamic environment of the AV, various studies have been proposed in the literature. Many researchers worked on different beam selection strategies for the AV. Also generating datasets that are spatial consistent and are time-evolving is important to assess the ML techniques. To complement the CNN, FL is further used in beam selection to reduce the complexity of the system.

Klautau et al. (2018) present a methodology for generating channel data in 5G mm-Wave scenarios. The goal is to make it easier to investigate ML- based problems related to the PHY of 5G mmWave MIMO. The proposed methodology simplifies the production of data in complex (and potentially realistic) mobility scenarios by continuously invoking a traffic simulator and a ray-tracing simulator. The generated datasets are highly valuable when spatial consistency and time evolution are required to evaluate the ML technique. In the current situation, creating propagation channel data is a realistic solution to reduce data scarcity while reaping the benefits of RT precision. For example, RT can handle 5G needs like spatial consistency, which classical stochastic modeling has struggled with. Simulated datasets actually complement, data from measurements, which can be used to develop and validate simulated data and statistical channel models when new information becomes available. Experiments with DL for beam- selection in vehicle-to-infrastructure (V2I) mmWave communications are also shown as actual examples of use for the obtained datasets. Because the current amount of data is limited, investigating the performance of specific DL architectures is beyond the scope of this research. Instead, the goal is to demonstrate the flexibility provided by the data generation methodology. This methodology can be used in applications other than V2I, as well as to generate datasets for ML problems such as classification, regression, clustering, and time-based sequence recognition.

Furthermore, (Klautau et al., 2019) proposed LiDAR-based beam selection in mm-Wave communication systems. Sensor data can be used to reduce the overhead of link configuration in mm-Wave communication systems. LiDAR is a high-resolution mapping and positioning sensor widely utilized in autonomous driving. They constructed a distributed architecture to reduce the overhead of the mm-Wave beam selection process. It assumes that the BS broadcasts its location over a low-frequency control channel and that the connected vehicle handles all processing. The vehicle estimates a set of M candidate beam pairings using its LiDAR data, its own position, and the broadcasted BS position, which are communicated to the BS via the control channel. The BS then trains the recommended beam pairs, and the best beam is selected for data transmission. They employed ML to tackle two major issues in the LIDAR-assisted mm-Wave system. Created a predictor to determine whether the channel is LoS or NLoS. Because beam selection is easier in the LoS setting, LoS detection is beneficial. Also, DL was employed with a neural network that is trained to perform top-M classification based on LoS and NLoS state estimates.

As suggested by (Mashhadi et al., 2021), FL can reduce the complexity. The transmission of LIDAR measurements from connected vehicles to the BS to assemble a centralized dataset for offline training would incur a significant communication overhead. Federated training helps in minimizing communication overhead.

The sensor data collected from AV can be shared using mm-Wave communication in 5G (Gonzalez-Prelcic et al., 2017). The BS selection is done for the AV's journey through the city.

Multihop cellular network is conventionally used as the strategy for BS selection (Marathe et al., 2008). Leveraging the side information like GPS coordinates, LiDAR data can reduce the communication overhead (Tran et al., 2019). LiDAR is a sensor mounted on the AVs used for obstacle detection and better beam selection in V2I communications for the LoS and NLoS transmissions (Klautau et al., 2019), (Hua et al., 2019).

2.1 Problem formulation

AVs can reduce fatal accidents significantly by up to 90% by eliminating driver error with the reduction in travel time as traffic congestion decreases and lane capacity increases (Litman, 2017). The sensor data collected from AV can be shared using mm-Wave communications which are considered a pre-eminent technology in 5G. Inspired by the wide- ranging application of ML, which includes image processing, finance, economics, and so on, it is projected as one of the most powerful technologies in 5G and beyond networks (Jiang et al., 2017), (Klaine et al., 2017). The problem statement revolves around intelligently selecting the best BS for AVs using sensor data in the urban area. Using communication information can be complex, time-consuming, and costly. Hence, leveraging the side information like GPS coordinates, and LiDAR data can reduce the communication overhead (Tran et al., 2019). Due to the short wavelength of mm-Wave and high directional beamforming, the MIMO systems are highly vulnerable to link blockage. Switching to an unblocked direction, and selecting the best BS is an effective solution to overcome blockage and restore communication links.

The ML-based communication systems have the potential to improve communication algorithms in terms of reliability, generality, latency, and energy efficiency. Modern ML techniques have recently achieved breakthroughs in many different domains along with communication systems (Yangli-ao Geng et al., 2019).

Inadequate system models: Signal processing algorithms in communication systems have rigid foundations in statistics and information theory. These algorithms are optimized for mathematically convenient models such as linear, stationary, and gaussian statistics, but not for real systems with many imperfections and non-linearities. ML-based communications system does not require a rigidly defined model for representation and transformation of information and can be easily optimized in an end-to-end manner for a real system with harsh realistic effects.

Parallelization gains of NNs: NNs are universal function approximators (Hornik et al., 1989). Since the execution of NNs can be highly parallelized using data, there is some hope that "learned" algorithms can be executed significantly faster and at a lower energy cost than manually "programmed" counterparts. Specialized hardware for ML applications. ML-based communication systems optimize end-to-end system performance.

Limiting functional block structure: Conventionally communications systems are represented through a chain of multiple independent processing blocks; each executing a well-defined and isolated function (e.g., coding, modulation, channel estimation, equalization). However, it is not clear that individually optimized processing blocks achieve the best possible end-to- end performance. In fact, we are introducing artificial barriers and constraints to efficiency. For example, we do not necessarily care how well we can estimate the channel with a given scheme, or how well anyone's independent function works, rather we seek to optimize end-to-end system metrics jointly with overall components. A learned end-to-end communications system will likely not possess such a well-defined block structure as it is trained to achieve only the best end-to-end performance.

The mm-Wave communication is the modern efficient tool for leveraging the sensor data to reduce the communication link configuration overhead. The intricacy and poor results obtained in beam selection using communication motivate us to use DL. The data set generation using ray- tracing techniques and a LiDAR sensor is described. Applied FL where the BS broadcasts its position to all nodes and uses LiDAR data as a dataset to predict the best station using DL technique.

CHAPTER 3

Background

A citywide map is imported from OpenStreetsMaps organization (OSM.org) in the form of OSM files and 3D world is created using Webots software using this map. Traffic is generated in this map using simulation for urban mobility (SUMO) software.

In this 3D world, 60 vehicles are taken into consideration where each vehicle has mounted LIDAR and GPS on the front, rear and at the center of the vehicle to get accurate positioning data out of it. LIDAR is a sensor mounted on the AVs for obstacle detection and better beam selection in V2I communications for the LoS and NLoS transmissions. GPS coordinates are used to get the longitudes and latitudes of each vehicle at different time instants.

Raytracing of this model is performed where 1 vehicle and 3 BSs around it is considered, where BS acts as a transmitter and vehicle acts as a receiver. Raytracing is performed using the SBR method using mm-wave communication. The strength of power beam is calculated among the 3 beams signals and the best base station is selected based on the highest beam power.

This outputs data generated by all the 60 vehicles at different time instants are Timestamp, Name, Model, GPS, speed, Lidar, BS and this is further used in our model.

Labels are generated out of ray tracing which is further compared with ML model to get the accuracy of the model.

Further, Federated learning algorithms are applied to this model which helps in minimizing communication overhead by training the model at the client side.

From previous works we get the system model and propagation model on which the data is being generated. In addition to that the CNN based inception model on which the federated learning models are being applied.

3.1 System model

For 5G mm-Wave MIMO channels, ray tracing is a promising simulation approach. It provides accurate results but the computational cost increases exponentially with the maximum allowed number of reflections and diffractions. Also, for good Ray tracing accuracy, the scenario should have detailed specifications (geometry, material, and size) of buildings, and vehicles which makes it a site-specific simulation (Klautau et al., 2018). 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, and SUMO is used as the traffic simulator, coupled with MATLAB to assess communication characteristics using accurate ray tracing. 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 a downtown model of Rosslyn, Virginia as it is heavily urbanized. Friis equation is used to find the ideal power received (P_{rx}) in dB at an antenna from basic information about the transmission and is given as

$$\boldsymbol{P_{rx} = P_{tx} + G_{tx} + L_t} \tag{1}$$

where $\boldsymbol{G_{tx}}$and $\boldsymbol{G_{rx}}$ are transmitted antenna gain and receive antenna gain respectively in dB.

P_{tx} is the power gain of the transmitting antenna in dB.

P_t is the total power loss in dB.

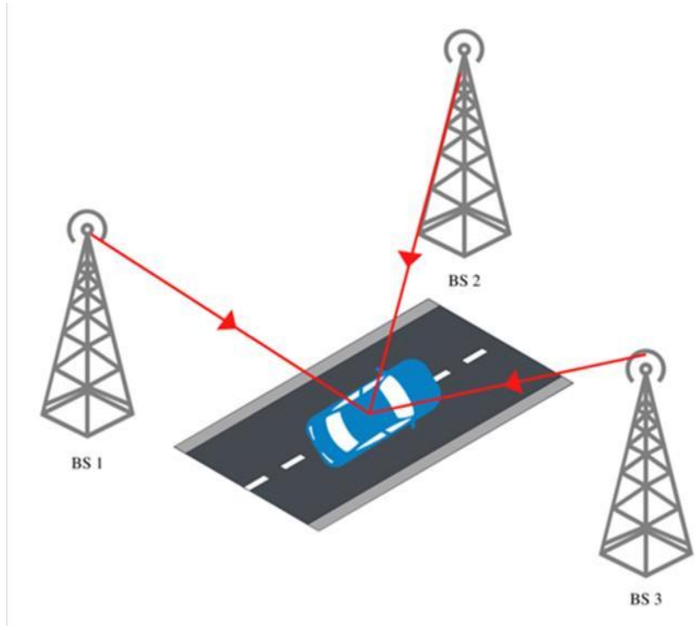

Figure 1: System model

3.2 Propagation model

Propagation factors such as reflection, scattering, diffraction, refraction, absorption, and atmospheric particles affect the transmitted signals in wireless communication. The propagation model facilitates the prediction of propagation loss and attenuation occurring in the signal traveling through the environment. The path loss includes free-space losses and reflection losses. The ray tracing model used in this simulation computes multiple propagation paths. The model learns the LOS path by launching rays from transmitter to receiver. If the ray does not interact with any surface before reaching the receiver, then it is a LOS transmission. The SBR method is used for NLOS transmission as the computational complexity increases linearly with the number of reflections while in the image method for NLOS, computational complexity increases exponentially with the number of reflections, which makes the SBR method faster than the image method. The model calculates losses using Fresnel equation for each reflection. In the SBR method, many rays are launched from the geodesic sphere as they are approximately uniformly spaced, centered at Tx. The method traces every ray from the Tx. The implementation used here considers only reflections. When the ray hits a flat surface, the ray reflects according to the law of reflection. When the ray hits the edge of a surface, the ray produces diffracting rays based on the law of diffraction. For every launched ray, the Rx is surrounded by a sphere, called a reception sphere, whose radius is proportional to the angular separation of the launched rays and the distance the ray travels. If the ray intersects the sphere, then the model considers the ray a valid path from Tx to Rx. The model calculates losses using a Fresnel equation for each reflection. The power losses include free-space path losses (FSPL) and reflection losses (RLs).

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

Reflection Loss (RL): As the ray interacts with the surface at some angle, and RL is calculated using Fresnel's equation. The Ray Tracing model computes RL by using the reflection matrix computations. For the current simulation, the materials are considered perfect reflectors; hence reflection loss is equal to zero.

Free-space path loss (FSPL): The FSPL in the far-field of the Tx in dB is given as follows:

$$\boldsymbol{FSPL = 20log((\frac{4\pi r}{\lambda}))}$$

(3)

where r is the distance between Tx and Rx antenna and λ is the wavelength. Although the mm-Wave signals experience higher attenuation in FSPL and shadowing, 5G networks use highly directional phased antenna arrays and beamforming technology to achieve sufficiently high antenna gains.

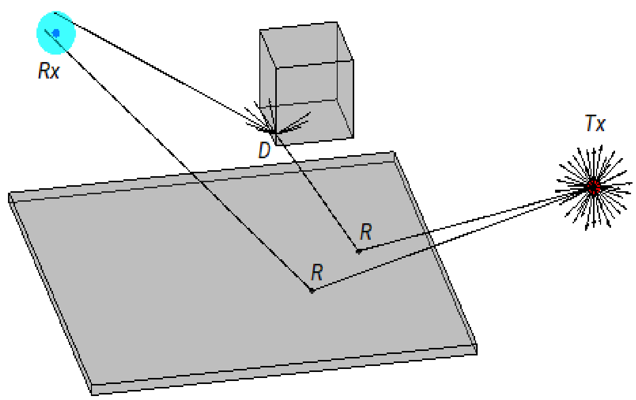

Figure 2 - SBR Method

3.3 Deep Learning

As a three- or more-layered neural network, DL is a subset of ML. DL enables systems to cluster data and produce incredibly precise predictions through a combination of data inputs, weights, and biases. These elements work together to efficiently identify, categorize, and describe things in data. The capacity of DL to manage large volumes of data has shown it to be a very beneficial technology. The dataset's train-valid-test split is a technique for evaluating the DL model's performance. The DL model uses the training dataset, which is a set of data, to figure out and fit the parameters. Validation dataset is a set of data that is used to give an unbiased evaluation of a model that has been fitted to the training dataset while optimizing the model hyperparameters. A test dataset is a collection of data that is used to offer an unbiased evaluation of a final model that has been fitted to the training dataset.

3.3.1 Convolutional Neural Network

CNN stands for convolutional neural network that is designed to approximate human vision and is a type of neural network. CNN falls under the category of DNN used to evaluate visual imagery in deep learning. It employs a technique known as convolution, which is a mathematical operation on two functions that yields a third function that expresses how the shape of one is influenced by the shape of the other. Multiple layers of artificial neurons make up CNNs. Artificial neurons are mathematical functions that calculate the weighted sum of various inputs and produce an activation value as a result. The basic structure of CNN is shown in Fig 2. When data is fed into a CNN, each layer generates several activation functions, which are then passed on to the next layer. Typically, the first layer extracts basic features. This information is passed on to the next layer, which is responsible for detecting more complicated features. It can detect even more complicated traits as we proceed further into the network. Feature extraction is the primary function of a convolutional layer. The output of the convolutional layer is then fed to the DNN for training. The classification layer generates a set of confidence scores (numbers between 0 and 1) based on the activation map of the final convolution layer, which indicates how likely the input is to belong to a class. The pooling layer is responsible for shrinking the convolved feature's spatial size. By lowering the size, the computational power required to process the data is reduced. Average pooling and max pooling are the two types of pooling.

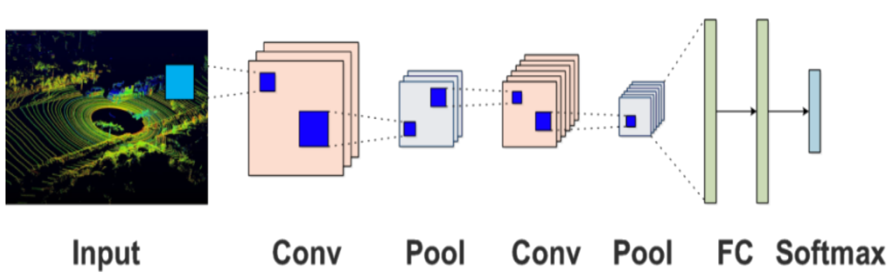

Figure 3 - General CNN architecture

3.3.2 Inception Model

The Inception V3 is a DL model for image classification that uses CNN, it was developed by a team at Google. It helps in avoiding overfitting when multiple deep learning layers are being used. The Inception model is made up of multiple Inception modules. Convolutions of various sizes are used to capture various sizes of information in the input. Inception has a lower computational cost than VGGNet or its higher-performing successors. This has allowed Inception networks to be used in big data scenarios, where large amounts of data must be processed at a low cost or where memory or processing power is fundamentally constrained, such as in mobile vision situations.

3.3.3 Deep Learning Model using LIDAR data

The proposed CNN architecture is given in Fig. 3 . The input to this CNN model is a feature map of [10,240,240], which is fed into the initial convolution layers. 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. It provides a novel architecture to reduce the computational cost while keeping the accuracy intact. The model contains four 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 that of the convolution output channel.

Dropouts are also added to our model for preventing overfitting with a value of 0.2 i.e., 20% of neurons are nullified towards the next layer and leaves unmodified all others.

Finally, there are another 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. Finally, the output of the linear layer is passed through a softmax layer to compute the given probabilities. The output vector is of dimension (1,3), denoting the probability of selection of the best BS out of the 3. To efficiently train the model, cross-entropy loss has been used coupled, shown in equation 5 with Adam Optimizer tuned with weight decay of 1×10^{-4}. and learning rate of 3.63×10^{-4}.

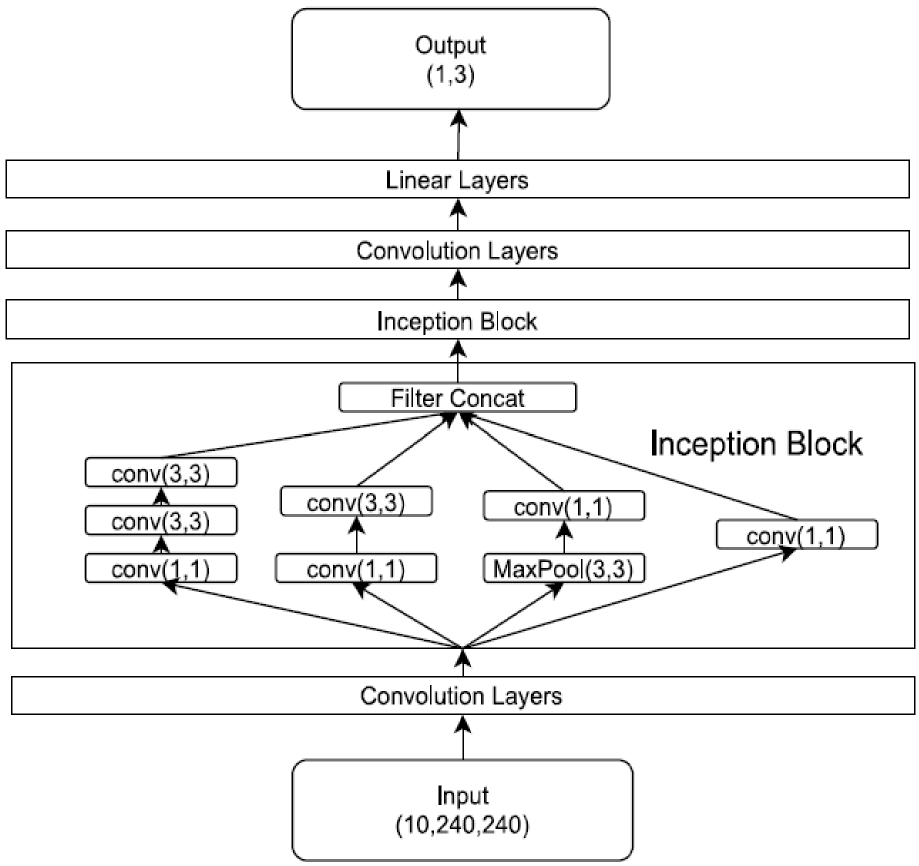

Figure 4 - LiDAR CNN model with Inception block.

3.4 Federated Learning

Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a ML problem, under the coordination of a central server. Each client's raw data is stored locally and not exchanged or transferred, instead the local clients and the central server only communicate through the model parameters to achieve the learning objective. So, the data stays on the devices only (here vehicles) and need not transferred to the central server.

It starts from the server side by either initializing the model randomly or the model is pre-trained on some data that is publicly available. A copy of this model is sent to the devices where it is updated using the data present on the device. After the local training on the devices, the updates are sent to the server. All the updates from the client side are aggregated on the server side to obtain an improved model. Now, in the second round, these updates are sent to the devices where the data is trained on this model and further these updates are again sent to the server.

This happens for ‘N’ communication rounds till the model can intelligently predict the desired output.

To summarize, federated learning enables clients (customer’s computing devices) to collaboratively learn a shared prediction model while keeping all the training data on device, decoupling the ability to do machine learning from the need to store the data in the cloud.

This approach stands in contrast to traditional centralized machine learning techniques where all the local datasets are uploaded to one server as well as to more classical decentralized approaches which often assume that local data samples are identically distributed.

Federated learning enables to build a common, robust machine learning model without sharing data, thus addressing critical issues such as data privacy, data security, communication overheads.

FL has become popular because of the following reasons, a) privacy of client’s data is preserved, b) the data size is reduced as models are shared, c) latency of the model is improved, and d) we get a better battery life.

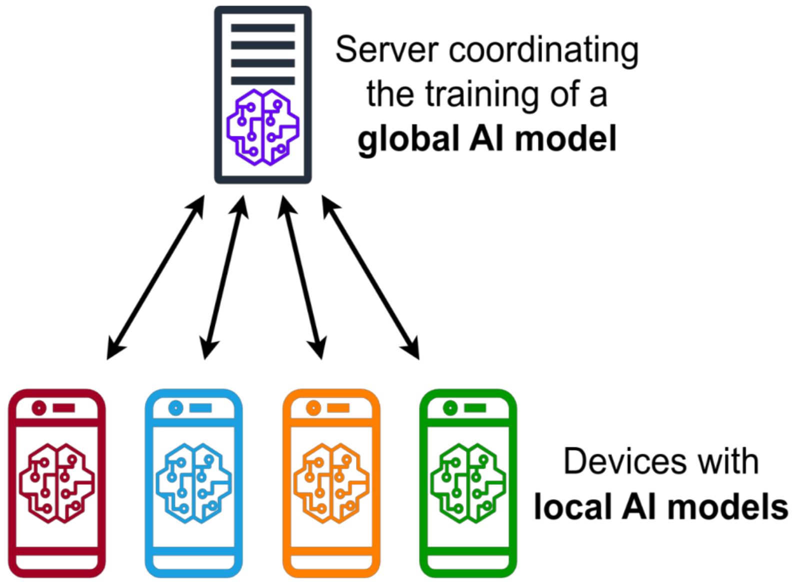

Figure 5 - Basic structure of Federated Learning

CHAPTER 4

Data Generation

We use an open-source robotics simulator, Webots, and SUMO as our traffic simulator, coupled with MATLAB, to assess communication characteristics using accurate time ray tracing. Fig. 2 shows the data generation process used to generate the data sets as used in the previous work.

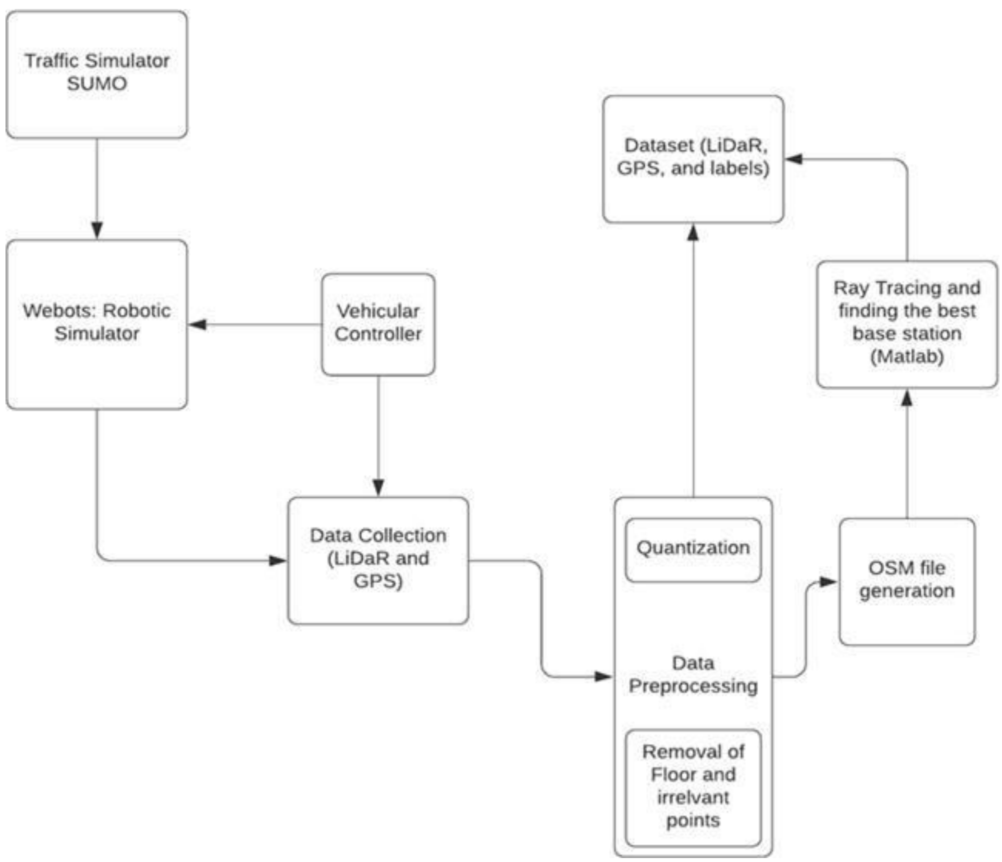

Figure 6 - Data Generation Process

4.1 Simulation

The map of the city is then used to generate the dataset. We used the data generation process as in Fig. 6. A portion of the city, which was to be used for simulation, was selected, and imported from openstreetmaps.org. Different locations are tested for BSs, they are compared according to the amount of LiDAR samples generated by different arrangements of BSs in the same simulation time. Three base stations are selected within a 100-meter radius of each other as in Fig. 1 according to system model pro- posed in the previous work. The base stations will be at an altitude of 5m above the ground. A 3D model of the world is generated in Webots and traffic is generated in the simulation using SUMO.

Webots is an open-source robotic simulator that simulate a 3D model of the city from the OSM files. Simulation for Urban Mobility (SUMO) is used to realistically simulate the traffic. The trips and the vehicles are generated randomly with SUMO.

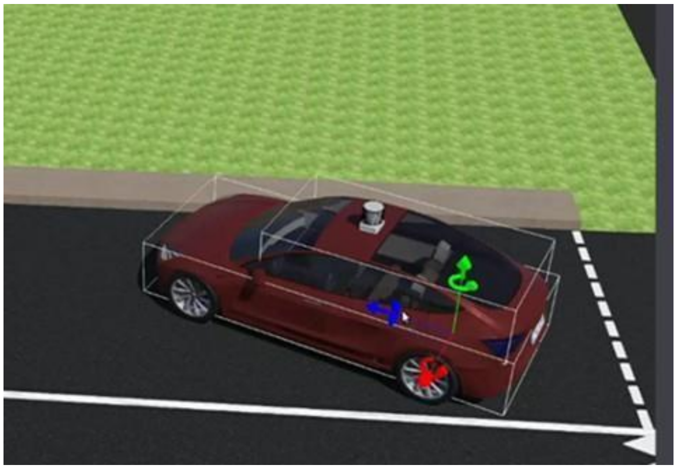

Figure 7 - A vehicle in the 3D model of the world

In Fig. 7, we can see the LIDAR sensor on the roof of the vehicle. Each vehicle consists of a LIDAR sensor and three GPS sensors that are simulated using standard LiDAR and GPS models in Webots. Velodyne HDL 64E is the LiDAR sensor used with a range of up to 120 meters. The LiDAR sensor was mounted on the vehicle's roof, and the GPS was mounted at the front, center, and rear for efficient retrieval of the vehicle's orientation. Standard GPS and LiDAR models are used are used to simulate the working of the 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. We also consider that GPS is mounted on the vehicle and is devoid of any noise or errors.

4.2 Preprocessing

Data collection is followed by data pre-processing, which includes quantization and the removal of irrelevant points. Data is quantized with step size of with step size of

1.0 in x-plane and z-plane, and 0.5 in y-plane. We also remove the points around the car in pre-processing. As the antenna is of height 5m in y-plane and its range is 120m in x-plane and z-plane, the quantization results in input array size of [10,240,240]. The total samples are then divided in the ratio of 8:1:1 for training, validation, and testing.

4.3 Ray Tracing

The next thing that must be done is MATLAB ray tracing, which finds the propagation path and losses effectively. This helps in choosing the best base station for a vehicle, thereby generating data labels.

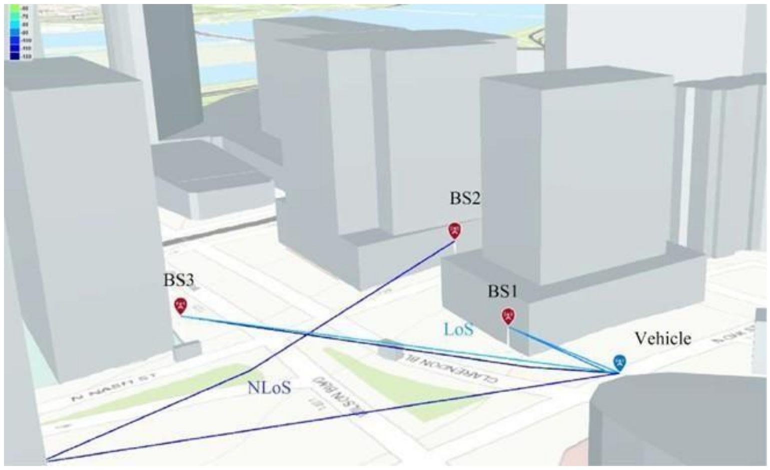

Figure 8 - Ray Tracing for LOS and NLOS communication between vehicle and base station

The transmitter taken is a 4 × 4 uniform rectangular arrays (URA) with element spacing of 0.1 meters in both X and Y directions. As shown in Fig. 8, the antenna is located at an altitude of 5m surface of the building or terrain with a transmitted frequency of 60GHz at 1W. The ray tracing model used in this simulation computes multiple propagation paths. The model learns the LoS path by launching rays from transmitter to receiver. If the ray does not interact with any surface before reaching the receiver, then it is a LoS transmission. If the rays interact with any surface, then it is NLOS path and SBR method is used. Ray tracing with the Shooting and Bouncing Rays (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.

4.4 Datasets

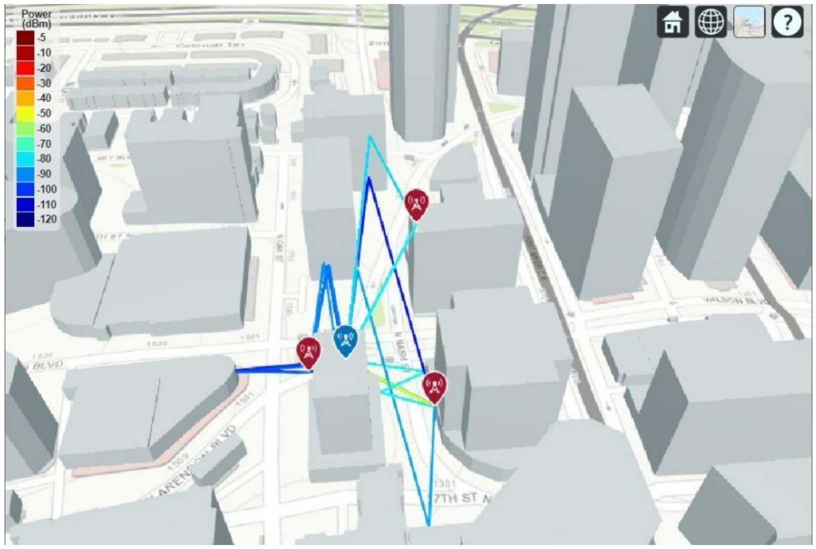

Figure 9 - Ray Tracing between vehicle and base station for Rosslyn

A 15k dataset of Rosslyn is used that is the previously created dataset. For bigger data set of 40k samples we have simulated a 3D world with traffic in Webots and SUMO to get the LIDAR data. The simulation time was increased to collect more samples. The location of the base stations was kept the same as the previous data set. Preprocessing is done on this data for quantization and removal of irrelevant points. Ray Tracing was done using MATLAB to generate data labels for the dataset as shown in Fig. 9. The rays of different colors show different intensity of power beams from base stations to vehicle.

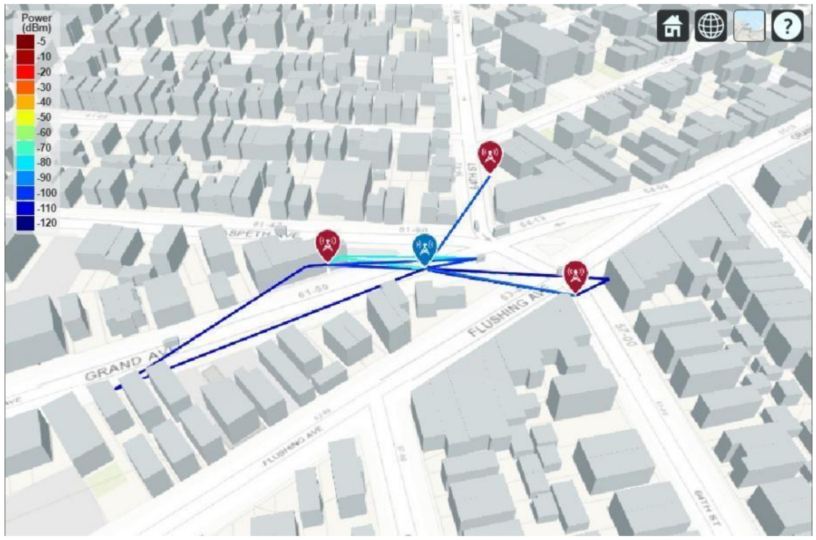

Figure 10 - Ray Tracing between vehicle and base station for NewYork city.

A portion of the city that has to simulated is selected and imported. Three base stations are selected that are within 100m radius of one another. Different locations in the city are tested for BSs but they are selected depending on the traffic at the selected location. Then a 3D world was simulated in Webots and SUMO to get the LIDAR data. Preprocessing is done on this data for quantization and removal of irrelevant points.

Ray Tracing was done using MATLAB to generate data labels for the dataset as shown in Fig. 10. For comparison between 2 cities, 15k samples of Rosslyn city & New York city are compare.

4.5 RSSI

In telecommunications, received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal.

RSSI value can be used as a measurement of how well a receiver can “hear” a signal from sender. The reporting range of RSSI value varies from 0 to -120 dBm (0 best, -120 worst).

In an IEEE 802.11 system, RSSI is the relative received signal strength in a wireless environment, in arbitrary units. RSSI is an indication of the power level being received by the receiving radio after the antenna and possible cable loss. Therefore, the greater the RSSI value, the stronger the signal. Thus, when an RSSI value is represented in a negative form (e.g. −100), the closer the value is to 0, the stronger the received signal has been.

Typically, RSSI is a measure of dBm, which is ten times the logarithm of the ratio of the power (P) at the receiving end and the reference power (Pref). Power at the receiving end is inversely proportional to the square of distance. Hence RSSI could potentially be used as an indicator of the distance at which the sending mote is located from the receiving mote. When data from many such neighboring motes are combined, the location of the sending mote can be judged with reasonable accuracy.

RSSI and distance have a relationship which is derived as follows. As mentioned previously, RSSI is defined as ten times the logarithm of the ratio of power of the received signal and a reference power (e.g., 1mW). i.e., RSSI α 10 log P/Pref. This would mean that RSSI α log P. It is known that power dissipates from a point source as it moves further out and the relationship between power and distance is that power is inversely proportional to the square of the distance travelled.

In other words, RSSI α log(1/distance^2). Simplifying this relationship further we can conclude that RSSI α (–log distance).

The distance in this equation is found using the haversine equation which considers the latitudes and longitudes of the position of vehicle and base station which is further used to find the RSSI values from each BS.

The distance in this equation is calculated using the haversine formula which considers the latitudes and longitudes of the vehicle and base stations.

Haversine formula can be expressed as follows:

$$\boldsymbol{haversine(\theta) = (sin\frac{\theta}{2})^2} \tag{4}$$

The central angle haversine can be computed as follows using equation 4 -

$$\boldsymbol{haversine(\frac{d}{r}) = haversine(lat_2 - lat_1) + cos(lat_1) * cos(lat_2) * haversine(long_2 - long_1)} \tag{5}$$

where lat_1 & lat_2 are the latitudes of the two points. $long_1$ & $long_2$ are the longitudes of the two points. d is the distance between the two coordinates, r is the radius of the earth.

CHAPTER 5

PROPOSED APPROACH

5.1 Deep learning model using LIDAR sensor data.

LiDAR gives more accurate realistic 3D data of the surroundings compared to GPS, so we apply the proposed CNN architecture on LiDAR data, as shown in Fig. 11.

The initial convolution layers feature a large 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. It provides a novel architecture to reduce the computational cost for the same accuracy. The proposed model contains four 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 that of the convolution output channel. Finally, there are other convolution layers (two (7, 7) and two (3, 3) kernels) to reduce the dimension of the model again, 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 of 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).

Also, it can be modeled based on the number of BSs.

Finally, the proposed model chooses the highest probability output as the 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).

To efficiently train the model, the cross-entropy loss has been used, with the 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 71.64% with 30 communication rounds and automatic mixed-precision set to 16 floating bits.

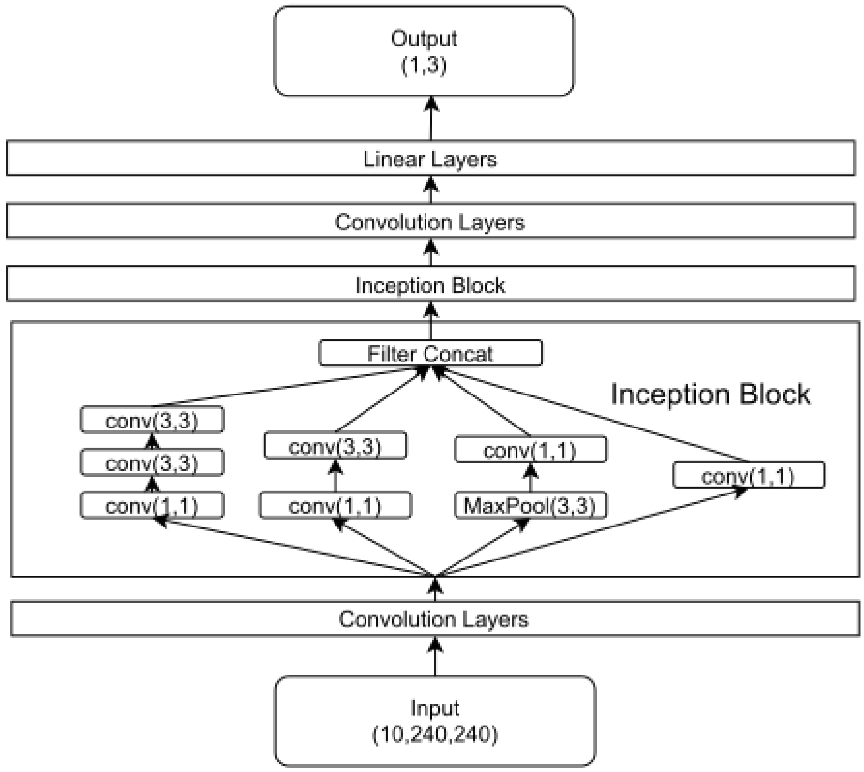

Figure 11 - The proposed LIDAR inception-based CNN model.

5.2 Federated learning on deep learning model

Using CNN on LiDAR achieves good accuracy for BS selection; however, it incurs a huge communication overhead during the transmission of data from the vehicle to BS while posing a significant security risk. Therefore, we propose to use FL, which 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. Also, we verified the results using FedProx (a modified version of FedAvg) and observed the same results. Let θ be the weights of the model used in training and V be the number of vehicles present, with N being the overall number of vehicles. Therefore, θ_i^v represents weights of the model allocated to the vehicle v at i communication round, where each communication round represents an aggregation of weights of different vehicles using average at the BSs and synchronizing them. Algorithm 1 represents the training loop for federated learning for n communication rounds between the BS and vehicles. FedProx, differ in line 6 in Algorithm 1, where BS selects a subset of vehicles and optimizes the loss function with the **proximal term** $\frac{\mu}{2}(\| \theta - \theta_t \|)^2$, where θ_t is global weights and θ is the client weights and μ is the scaling factor.

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. 13. 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 the updates. We train the CNN classifier using the cross- entropy loss function with an Adam optimizer with an initial learning rate of 1×10^{-3}. and a batch size of 64. The models are trained for three epochs per data set, with 10 communication rounds.

Algorithm 1: FedAvg for LiDAR-assisted BS selection

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

Figure 12 - FL Algorithm of our model

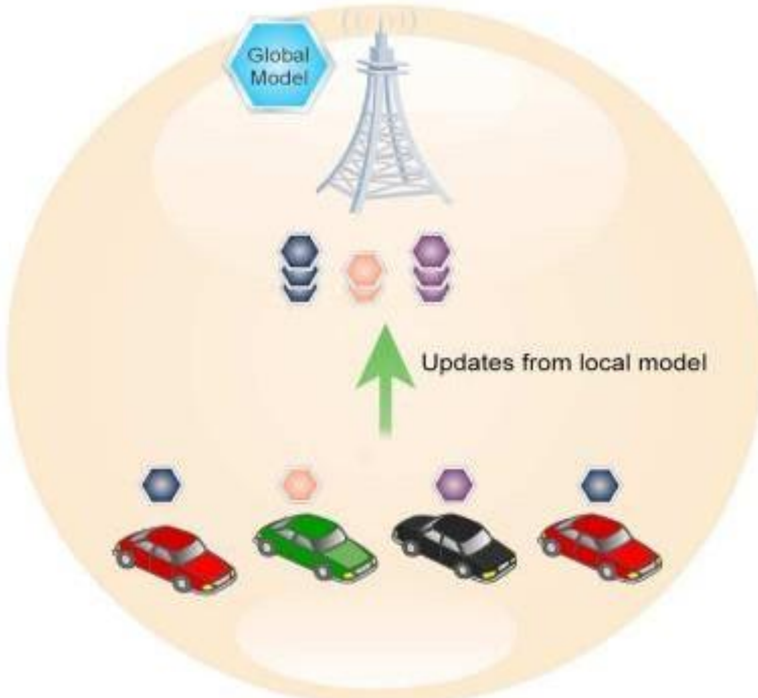

Figure 13 - Process of FL

5.2.1 Federated Learning Algorithms on our model

Using CNN on LiDAR data achieves good accuracy for BS selection, however it incurs a huge communication overhead during the transmission of data from the vehicle to BS while also posing a significant security risk. Thereby, we propose to use FL which aims to predict a realistic model which accounts for the local data without sharing it with the server. In FL, global and client models are defined and are synchronized. A copy of global model is sent to the clients where the model is updated using the data on each device. This is done by performing stochastic gradient descent (SGD) for 'e' epochs on the client side. After the local training is done on the clients, the updates are sent to the server. All the updates from the client side are aggregated and the process continues for 'n' communication rounds till an improved model is obtained. On the server side, after improving the model the accuracy of the model is checked. The vehicles use federated averaging (FedAvg), In fedavg, each client has its own data which is being identically distributed. And so, the number of local epochs considered is the same for all the clients. Also, another algorithm known as fedProx has been applied on the same model to observe different set of results. In fedprox, we generalize fedavg by allowing for variable amounts of work to be performed locally across clients based on their available systems resources. proximal term is added to the local subproblem to effectively limit the impact of variable local updates.

Let θ be the weights of the model used in training and V be the number of vehicles present, with N being the overall number of vehicles. Therefore, θi represents weights of the model allocated to the vehicle V at i communication round, where each communication round represents an aggregation of weights of different vehicles using average at the BSs and synchronizing them. Algorithm 1 represents the training loop for federated learning for n communication rounds between the BS and vehicles. FedProx, differ in line 6 in Algorithm 1, where BS selects a subset of vehicles and optimizes the loss function with the **proximal term,**

$\frac{\mu}{2}(\| \theta - \theta_t \|)^2$, as shown in the Algorithm 2,

where where θ_t is global weights and θ is the client weights and μ is the scaling factor.

.

Algorithm 2 `FedProx` (Proposed Framework)

Input: K, T, μ, γ, w^0, N, p_k, $k = 1, \cdots, N$

for $t = 0, \cdots, T - 1$ **do**

Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)

Server sends w^t to all chosen devices

Each chosen device $k \in S_t$ finds a w_k^{t+1} which is a γ_k^t-inexact minimizer of: $w_k^{t+1} \approx \arg\min_w \; h_k(w;\, w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2$

Each device $k \in S_t$ sends w_k^{t+1} back to the server

Server aggregates the w's as $w^{t+1} = \frac{1}{K}\sum_{k \in S_t} w_k^{t+1}$

end for

Figure 14 - FedProx Algorithm

CHAPTER 6

Results and Discussions

Federated learning algorithms are applied of 4 datasets i.e., 15k samples of Rosslyn city, 15k samples of New York city and 40k samples of Rosslyn city and 40k samples of New York city.

Hyper parameter tuning is performed to get a comparison between different parameters.

Following are the comparisons on our model for Rosslyn city of 40k samples–

1) Comparison based on number of **inception layers** on our CNN model for 30 communication rounds on fedprox algorithm.

1.A) 2 inception layers - The accuracy is 71%.

1.B) 3 inception layers - The accuracy is 71.16%.

1.C) 4 inception layers - The accuracy is 70.93%.

The below graph shows the comparison between different inception layers applied to our CNN model.

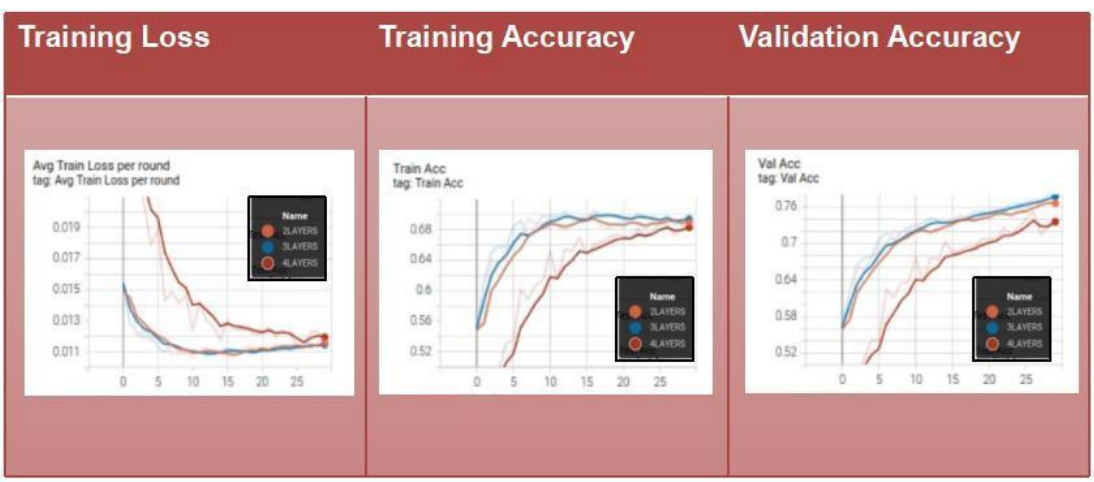

Table 1 - Comparison based on number of inception layers

2) Comparison based on **number of communication rounds** applied to our federated learning algorithms (Fedavg & Fedprox). Here the inception layers are fixed to 4.

2.1) Fedprox Algorithm

2.1.A) For 20 communication rounds the accuracy is 68.98%.

2.1.B) For 30 communication rounds the accuracy is 70.93%.

2.1.C) For 40 communication rounds the accuracy is 70.39%.

Table 2 - Comparison based on the number of communication rounds (FedProx).

2.2) Fedavg Algorithm

2.2.A) For 20 communication rounds the accuracy is 68%.

2.2.B) For 30 communication rounds the accuracy is 70.9%.

2.2.C) For 40 communication rounds the accuracy is 68.98%.

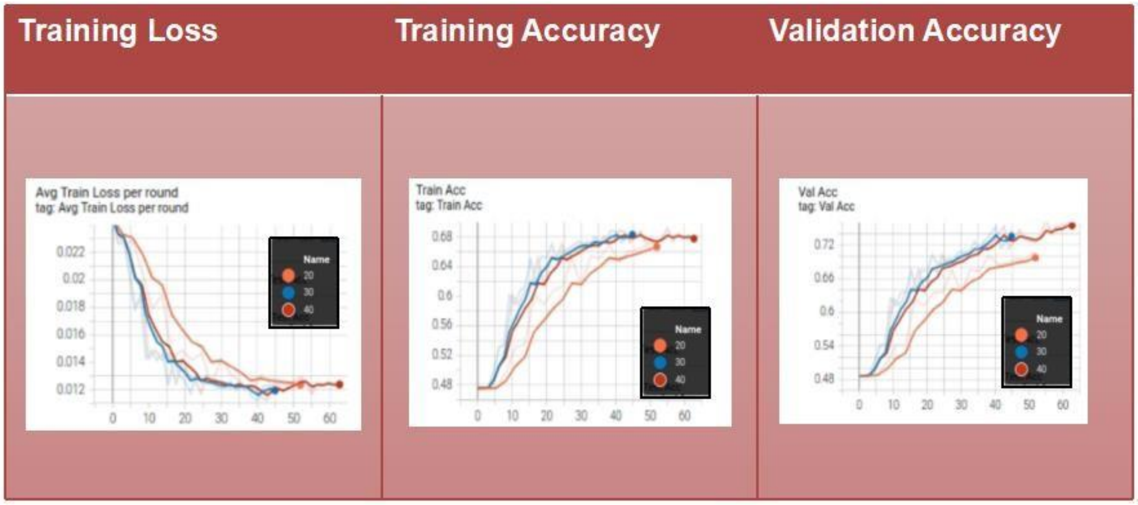

Table 3 - Comparison based on the number of communication rounds (FedAvg).

3) Comparison based on **different federated learning algorithms** for 15k dataset and 40k dataset of Rosslyn city.

	FedSGD – test accuracy	FedAVG – test accuracy	FedPROX – test accuracy
15K samples	58.91%	60%	60.3125%
40K samples	68.65	68.98%	70.39%

Table 4 - Comparison based on the algorithm used.

4) Comparison based on **two different cities** i.e., Rosslyn and New York for 15k samples of data –

Table 5 - Comparison based on different cities for 15k samples.

Communication rounds	Rossyln city - Test Accuracy	New York city - Test Accuracy
10	60.3125%	64.06%
20	60.703125%	63.91%
30	60.5%	62.2%

5) Comparison based on **different optimizers** used on fedavg algorithm and fedprox algorithm –

OPTIMIZERS	FEDAVG – test accuracy	FEDPROX – test accuracy
1) Adam	60%	60.3125%
2) Adamax	59.90%	58.98%
3) AdamW	59.00%	60.7%
4) RMSprop	59.95%	62.109%
5) AdaGrad	52.03%	59.375%
6) SGD	52.00%	46.484%

Table 6 - Comparison based on different optimizer.

6) Comparison based on **different cities** used on fedavg algorithm and fedprox algorithm for **40k samples** -

A) Fedprox

communication rounds	Rossyln city - Test Accuracy	NewYork city - Test Accuracy
10	68.56%	72.01%
20	68.96%	72.89%
30	70.93%	73.99%

Table 7 - Comparison based on different cities for fedprox algorithm.

Table 8 - Comparison based on different cities of 40k samples.

B) Fedavg

communication rounds	Rossyln city - Test Accuracy	NewYork city - Test Accuracy
10	69.20%	71.78%
20	68.89%	72.09%
30	70.55%	72.88%

Table 9 - Comparison based on different cities for fedavg algorithm.

Table 10 - Comparison based on different cities of 40k samples.

7) RSSI Results

RSSI values from the three base stations to the vehicle is calculated using the given formula using GPS coordinates.

$$RSSI= P_O-(20*log10(4*pi*distance/wavelength)) \tag{6}$$

where Po is an empirical constant and it's value is set to 31.0 dBm.

and distance is calculated using the haversine formula mentioned in equation 4 and 5, where the latitudes and longitudes of the vehicle and base station is taken into consideration for calculating the distance.

The accuracy based on RSSI values of the three base stations with respect to every vehicle on the model has occurred to be 37%.

Chapter 7

Conclusion and Future work

This work presents a methodology to realistically and accurately simulate the data for BS selection. 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.

Two FL algorithms are applied on this system model and compared with variations in parameters like the number of communication rounds, change in the number of inception layers, change in location, and the training data size. It shows some outstanding results which has been depicted in the tables.

The existing model will be strengthened by increasing the complexity of the communication environment and introducing techniques such as beam selection and beamforming as part of 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. 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.

[3] 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.

[4] 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.

[5] A. Klautau, N. Gonzalez-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.

[6] A. Klautau, P. Batista, N. Gonzalez-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] H. White, “Consumer data privacy in a networked world: A framework for protecting a privacy and promoting innovation in the globaeconom,” https://obamawhitehouse.archives.gov/sites/default/files/privacyfinal.pdf.

[8] 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.

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

[10] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner, “Sumo (simula- ¨ tion of urban mobility)-an open-source traffic simulation,” in Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002), 2002, pp. 183–187.

[11] Z. Yun and M. F. Iskander, “Ray tracing for radio propagation modeling: Principles and

applications," IEEE Access, vol. 3, pp. 1089–1100, 2015.

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

[13] 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.

[14] N. Gonzalez-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.

[15] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flotter ¨ od, ¨ R. Hilbrich, L. Lucken, 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/