LiDAR Data-based Base Station Selection using Deep Learning

M.Tech. Thesis

By

BHAGYASHREE GOUR

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2022

LiDAR Data-based Base Station Selection using Deep Learning

A THESIS

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

of

Master of Technology

by

BHAGYASHREE GOUR

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2022

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **LiDAR Data-based Base Station Selection using Deep Learning** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DISCIPLINE OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July 2020 to June 2022 under the supervision of Dr. Vimal Bhatia, Professor, IIT Indore.

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.

31/5/2022

Signature of the student with date
(NAME OF THE M.TECH. STUDENT)

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

31.05.2022

Signature of the Supervisor of
M.Tech. thesis (with date)
(Prof. Vimal Bhatia)

BHAGYASHREE GOUR has successfully given her M.Tech. Oral Examination held on **06/06/2022**.

31.05.2022

Signature(s) of Supervisor(s) of M.Tech. thesis
Date: 06/06/2022

Convener, DPGC
Date: 06/06/2022

Signature of PSPC Member #1
Date: 06/06/2022

Somnath Dey

Signature of PSPC Member #1
Date: 06/06/2022

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Dr. Vimal Bhatia for constant support, motivation and guidance through my project work.

I would like to express my gratitude to my PSPC committee members Dr. Santosh Kumar Vishvakarma and Dr. Somnath Dey, for their valuable suggestions and insightful comments. Their questions and suggestions helped me to widen my knowledge from various perspectives.

I would also like to thank the Department of Electrical Engineering, IIT Indore for support and for providing the required lab facilities.

I would also like to thank all the faculty members of Electrical Engineering Department for teaching me during my course work which led to strong foundation for this work.

I am grateful to my friends Siddhesh, Sresthha, Raj, Shubham, Mohit and PhD Scholar Puneet Singh for support and help throughout the project.

I am also greatly thankful to my family for their true support.

Bhagyashree Gour
M.Tech (Communication and Signal Processing)
2002102001
Department of Electrical Engineering, IIT Indore

Abstract

BS selection plays an important role in establishing communication links in the mm-Wave-based communication systems. For selecting the best BS, each BS performs a handshake with the UE. This selection is further challenging due to the increased communication overhead imposed by the handshake between multiple BSs and UEs. In this research, we investigate BS selection for AVs using 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 can be mitigated by DL-based communication systems. Further, FL is applied where BS broadcasts its position to all nodes to reduce communication overhead. The dataset generation is described using the RT technique and LiDAR sensor. Finally, the simulation results verify that the proposed algorithm performs considerably better, with an accuracy of 1.7 times better than 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 TL for variation in the number of samples for training, the city used for simulation, and the number of BSs. Furthermore, hyperparameter tuning is performed using PSO on CNN utilizing LiDAR data.

LIST OF PUBLICATIONS

Under Review

Mohit Mehta, Bhagyashree Gour, Vimal Bhatia and Nandana Rajatheva, "Federated Learning-based Base Station Selection using LiDAR Data," IEEE Communication Letters.

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
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
PSO	Particle Swarm Optimization
RT	Ray Tracing
Rx	Receiver
SBR	Shooting and Bouncing Rays
SUMO	Simulation for Urban Mobility
TL	Transfer Learning
Tx	Transmitter
UE	User Equipment
URA	Uniform Rectangular Array
V2I	Vehicle-to-Infrastructure

CHAPTER 1

Introduction

There is rapid development in the sector of AVs and their communication, popularly known as V2I communication (Litman, 2017). There is a prediction that AVs can reduce the number of accidents significantly. Attentive and economically driven AVs can also reduce fuel consumption, thus helping in minimizing the transportation cost. It helps in the conservation of other parts of the vehicle, and it reduces travel time with efficient parking and affordable taxis (Litman, 2017). AVs are equipped with multiple sensors and devices for ensuring safety. These sensors and devices collect large quantities of data while interacting with the environment for monitoring the movement of AV and ensuring safety.

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 technology in 5G and beyond the network. ML is one of the most promising technologies that has been proven conducive to solving numerous problems of telecommunications, like physical layer optimizations, network management, and conceives to support smart.

A novel real-time method for selecting the best BS (top 1 accuracy) from a number of BSs that are within 100 meters range from the AV while keeping track of traffic. We propose a DNN architecture along with the corresponding data preprocessing (LiDAR and GPS) technique for data-driven mm-Wave beam selection. 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. Further, we also proposed a model that works on the FL technique with the same dataset for predicting the best base station. With the rise of interest in preserving the privacy of data (Anonymous, 2013), the emerging technique FL helps in fortifying user privacy and takes advantage of user participation, also known as collaborative learning, 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 (Nguyen et al., 2019). With recent improvements in edge computing, FL may now be easily deployed in real-time, allowing for unprecedented large-scale flexible data collection and model training. The suggested model's universality was further tested by applying approaches such as TL for modification in the number of samples for training from 12000 to 24000 and 36000, the city used for simulation was chicago, and the number of BSs are increased from 3 to 5. Furthermore, PSO on CNN is used to tune hyperparameters using LiDAR data.

1.1 Thesis Outline

This chapter 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 detail about the fundamentals used further in the thesis. Section 3.1 discuss the system model used and Section 3.2 covers all the fundamentals of DL
- 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 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 method for generating channel data in 5G mm-Wave scenarios. The goal is to make it easier to investigate ML-based challenges related to the physical layer of 5G MIMO. By concurrently engaging a traffic simulator and a ray-tracing simulator, the suggested technique facilitates the creation of data in complicated mobility scenarios. In the current situation, creating propagation data is a realistic solution to reduce data shortage 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 validate and develop statistical channel models and simulated data when new information becomes available. The purpose is to show how the data generating process may be flexible. This technology may be used to challenges other than V2I, like clustering, classification, etc., as well as to produce datasets for ML challenges.

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. To decrease the cost of the mm-Wave beam selection process, they built a distributed architecture. 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 beam pairings using includes LiDAR information, its coordinates, and the BS location of the transmissions, 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.

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).
B. Mishra et al., 2020, suggest that CNN can tune its hyperparameters using PSO for getting the optimal performance in terms of accuracy by

changing a set of parameters. Recent research suggests that PSO can be used to select hyperparameters in CNN (Lorenzo et al., 2017). It helps to reach the optimal accuracy for suggested CNN architecture. The existing work holds a selection of the best beam pair and beamforming for single BS. The originality of this study is that it builds on previous work by selecting the best BS for AV from a pool of many options.

2.2 Problem formulation

AVs can reduce fatal accidents significantly by up to 90% by eliminating the 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 technology 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).

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: The fundamentals of signal analysis in communication systems are based on 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). Data may be used to highly parallelize the execution of NNs., 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 ambiguous that individually optimized learning blocks deliver the best possible results. 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

3.1 System model

For 5G mm-Wave MIMO channels, RT is a promising simulation approach. The number of reflections and diffractions in RT accounts for the computational cost. 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 Rossyln, 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

$$P_{\mathrm{rx}} = P_{\mathrm{tx}} + G_{\mathrm{tx}} + L_{\mathrm{t}} \quad (1)$$

where G_{tx} and 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. L_{t} is the total power loss in dB.

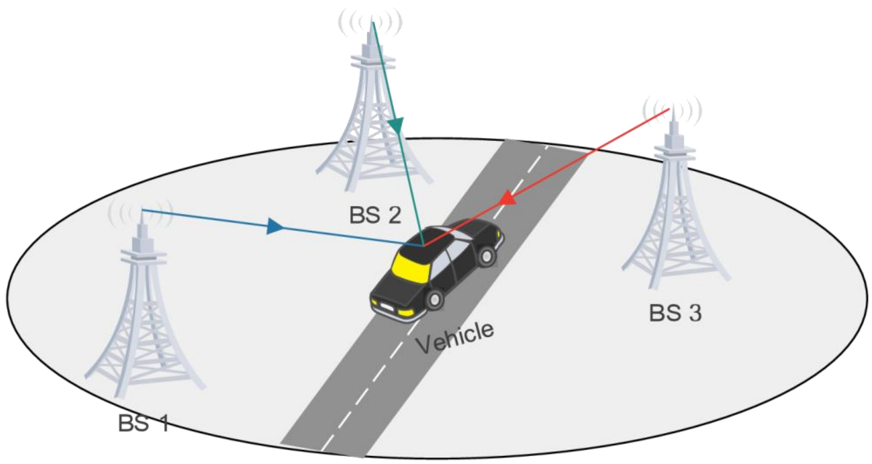

Figure 1: System model

Propagation model: The 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 (2). The ray tracing model used in this simulation computes multiple propagation paths (Schaubach et al., n.d.), (Yun & Iskander, 2015). 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 because the computational cost of the SBR method rises linearly with the increasing number of reflections, whereas the computational cost of the image method for NLoS increases exponentially with the number of reflections, making the SBR method efficient than the image method (Schaubach et al., n.d.). The model calculates losses using a fresnel equation for each reflection. In the SBR method, many rays are launched from the geodesic sphere as they are uniformly separated, centered at Tx. The method traces all the Tx rays. 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 reflection losses are calculated using the horizontal and vertical polarizations of signals through the propagation path.

$$L_{\mathrm{t}} = L_{\mathrm{fs}} + L_{\mathrm{r}} \tag{2}$$

Reflection Loss (L_{r}): As the ray interacts with the surface at some angle, and L_{r} is calculated using Fresnel's equation. The RT model computes L_{r} 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 (L_{fs}): The L_{fs} in the far-field of the Tx in dB is given as follows:

$$L_{\mathrm{fs}} = 20 \log\left(\left({}^{4\pi r}/_{\lambda}\right)\right) \tag{3}$$

where R is the distance between Tx and Rx antenna, and is the wavelength. Although the mm-Wave signals experience higher attenuation in L_{fs} and shadowing (Lili Wei et al., 2014), 5G networks use highly directional phased antenna arrays and beamforming technology to achieve sufficiently high antenna gains.

3.2 Deep Learning

DL is a subset of ML that is essentially a three-or more-layered neural network. DL seeks to emulate the human brain, while it falls far short of its capabilities, allowing systems to cluster data and generate extremely accurate predictions through a combination of data inputs, weights, and biases. These components collaborate to effectively recognize, classify, and characterize objects in data. Because of its ability to handle massive volumes of data, DL has proven to be a very useful technology. While a single-layer neural network can still produce approximate predictions, additional hidden layers can assist optimize and tuning for accuracy. DNNs are comprised of multiple layers of interconnected nodes, with each layer improving and optimizing the prediction or categorization. Forward propagation refers to the progression of calculations via the network. The visible layers of a DNN are the input and output layers. The DL model ingests data for processing in the input layer, and the final prediction or classification is performed in the output layer. Backpropagation is a method of training a model that uses methods such as gradient descent to calculate prediction errors and then modifies the weights and biases of the function by moving backward through the layers. Forward propagation and backpropagation work together to allow a neural network to make predictions and fix any faults. The train-valid-test split of the dataset is a method for assessing the performance of the DL model. The training dataset is a collection of data that is utilized by the DL model to learn 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.

3.2.1 Convolutional Neural Network

Around the 1980s, CNNs were developed and used for the first time. A CNN is a form of neural network that is designed to approximate human vision. A 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 a number of 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.

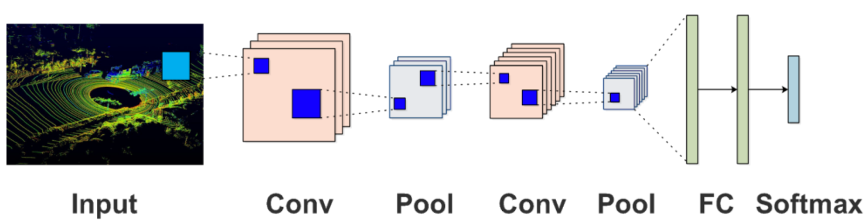

Figure 2: General CNN architecture

3.2.2 Inception Model

The Inception V3 is a DL model for image classification that uses CNN, it was developed by a team at Google. When multiple deep layers of convolutions were used in a model it resulted in the overfitting of the data. To avoid this from happening the inception model uses the idea of using multiple filters of different sizes on the same level. Thus, instead of having deep layers in the inception models, we have parallel layers, making the model wider rather than deeper. 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. In comparison to prior models and contemporaries, the inception V3 model has an extraordinarily low error rate (Szegedy et al., 2016a).

3.2.3 Transfer Learning

The technique of taking a pre-trained neural network and adapting it to a new dataset by transferring or repurposing the learned features is known as transfer learning. A model trained on a specific architecture is considered, and the learned weight from that model is used to initiate the training and classification of a completely new dataset. With a restricted computational resource, transfer learning is especially beneficial. Even when trained on extremely powerful GPU processors, many state-of-the-art models require many days or weeks to train. As a result, transfer learning allows us to use pre-trained weights as a starting point to avoid repeating the same process over and over again. Transfer learning frequently entails taking the pre-trained weights in the first layers, which are generally common to multiple datasets, and randomly initializing and training the remaining layers for classification purposes.

Learning or backpropagation happens only at the last layers in the transfer learning approach, which are initialized with random weights. Meanwhile, there are numerous techniques to transfer learning, and which one we apply is determined by the nature of the new dataset we wish to classify in relation to the pre-trained models' dataset.

The various CNN architectures analyzed over LiDAR data using TL are GoogLeNet and ResNet. GoogLeNet is a 22-layer (27 layers including pooling layers) CNN developed by Google researchers as a variation of the Inception Network. The GoogLeNet achieves efficiency by reducing the input while maintaining critical spatial information. ResNet, an architecture presented by Microsoft Research in 2015, established a new architecture called Residual Network. The ResNet is made up of numerous different types of residual blocks. However, depending on the architecture of residual networks, the operations in the residual block can vary.

3.3 Particle Swarm Optimization

Eberhart and Kennedy proposed PSO in 1995 as a population-based stochastic optimization approach for simulating the swarm behaviour of school of fish and flock of birds when hunting. Each group member (i.e., the particle) modifies its search mode by learning from all the candidate’s experiences. PSO is a heuristic search strategy that attempts to mimic the movements of a flock of birds looking for food. It is built on a population of particles travelling in search space with a particular velocity and position for each particle (Mishra & Sengupta, 2014).

The PSO technique starts with a collection of particles that are uniformly distributed across the search space. The locations of particles calculate the fitness function at each step of the iterations. If the current result outperforms previous results, the particle with the best result is noted, and the other particles should keep track of their own personal best results. The calculation of updated velocity, v_{id}^{t}, consists of three elements: inertia, local search, and global search. Then, using the position updating process given by (4) and (5), all of the particles are displaced according to their previous positions.

$$v_{id}^{t} = m * v_{id}^{t-1} + c_1 r_1 * (pBest_{id} - p_{id}^{t-1}) + c_2 r_2 * (gBest_d - p_{id}^{t-1}) \tag{4}$$

$$p_{id}^{t} = p_{id}^{t-1} + v_{id}^{t} \tag{5}$$

In (4), m denotes non-negative inertia weight, c_1 and c_2 are the acceleration, i represents the index of the particle and t is the iteration counter, $gBest_d$ denotes the fitness of the global optimum particle in the d^{th} dimension, and $pBest_d$ denotes the fitness of the local optimum particle in the d^{th} dimension. Constants c_2,r_2 and m,c_1,r_1, are factors regulating the impact of three elements on the result, which means that three values of current speed are modifiable in different applications. There are two common termination conditions for the algorithm: a maximum number of iterations and a sufficiently good fitness value.

CHAPTER 4

Data Generation

4.1 Generating the required environment

As an example implementation, we consider the downtown of Rossyln, Virginia, for generating the dataset and pre-processing methodology as shown in Fig. 3 (Klautau et al., 2018). The city-wide map has been imported from OpenStreetMaps (*Index Of /*, n.d.) in the form of OSM files, and the 3D world is constructed using Webots internal importer (*Webots: Robot Simulator*, n.d.). After creating the 3D map, the traffic is generated using SUMO (Krajzewicz, 2010). The distribution of surrounding vehicles follows Gaussian distribution over the whole dataset (training, validation, and testing) as shown in Fig. 4 (Abuelenin & Abul-Magd, 2014).

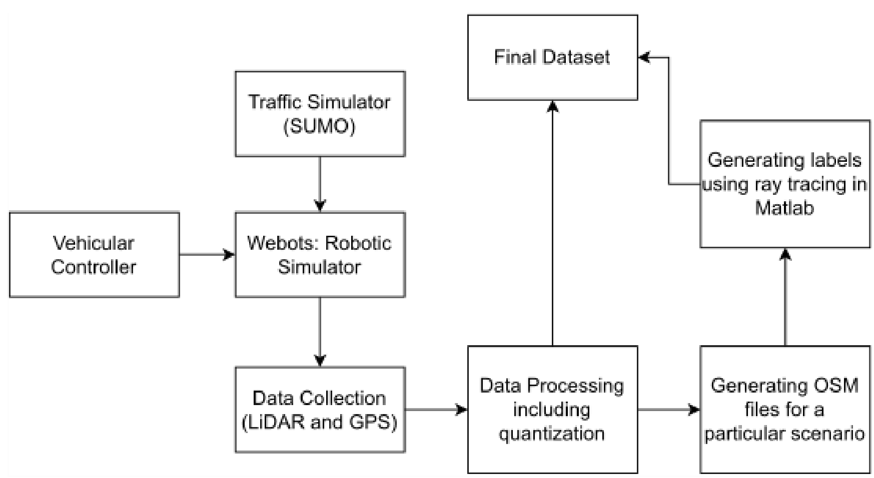

Figure 3: Data generation process

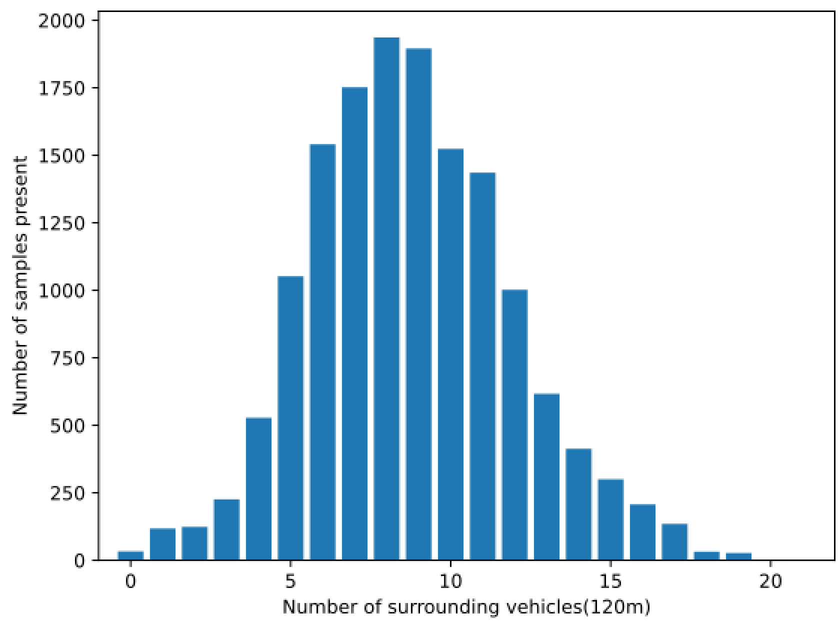

Figure 4: Distribution of surrounding vehicles over the collected data

4.2 Generating dataset using Ray Tracing and LiDAR sensor

We consider 60 vehicles in the 3D model, with each vehicle consisting of a LiDAR sensor - Velodyne 64E and three GPS sensors (Qin et al., 2021). 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 (*Webots: Robot Simulator*, n.d.). We also consider that GPS is mounted on the vehicle and is devoid of any noise or errors (*Webots: Robot Simulator*, n.d.). 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.

4.2.1 Ray Tracing

RT is used to gauge the propagation path and the losses accurately. The transmitter taken is a 4 × 4 uniform rectangular array (URA) with element spacing of 0.1 meters in both X and Y directions. As shown in Fig. 5, the antenna is located at an altitude of 5m surface of the building or terrain with a transmitted frequency of 60GHz at 1W (Klautau et al., 2018). RT 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.

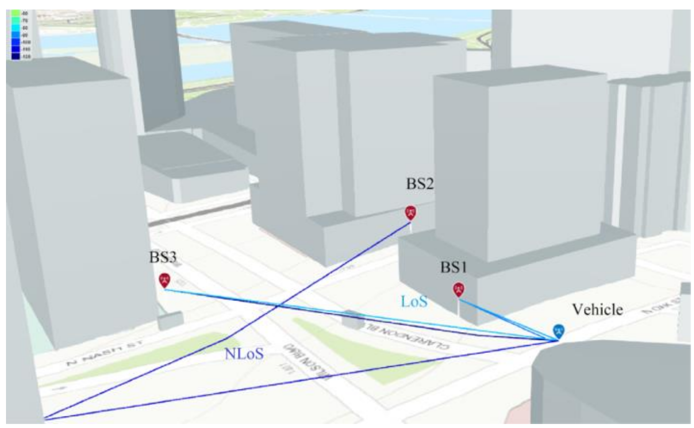

Figure 5: RT in MATLAB

4.2.2 LIDAR data

The LiDAR dataset is quantized following the parameter described in (Klautau et al., 2018) with quantization steps of 1.0 in the x-plane, 0.5 in the y-plane, and 1.0 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., 120m and y range is taken to be 5m.

CHAPTER 5

Proposed Approach

5.1 Deep learning model using LiDAR sensor data

A CNN is trained from scratch, this is used as baseline model to compare with other proposed models. The model uses 3 convolutional layers followed with a FC network. A simple architecture is used to compare and verify the trade-off between model complexity and performance. The architecture comprises of convolution blocks, consisting of convolution operation followed by ReLU as activation function and finally max pooling operation as shown in Fig. 6. After feature extraction, the feature maps are fed to the 3 layer FC network having 128, 64, 3 nodes respectively. Finally the softmax classifier is used to determine the likelihood of the input belonging to a class.

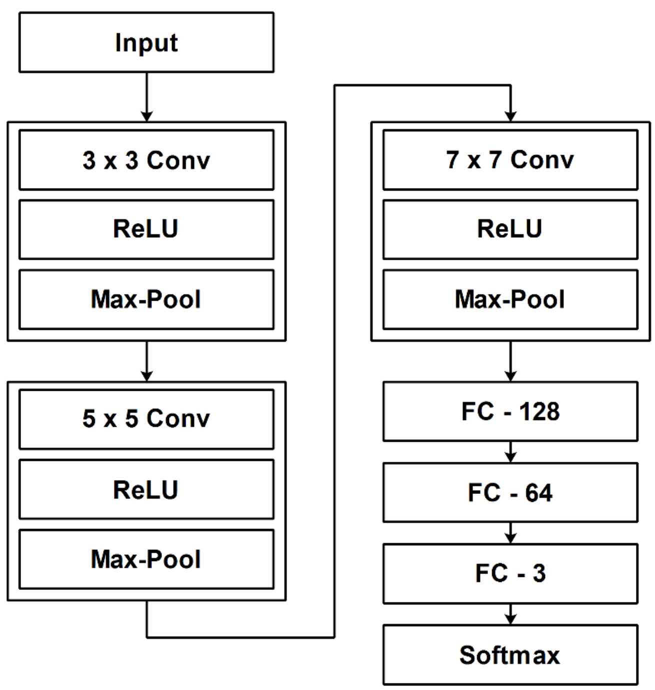

Figure 6: Baseline CNN model

The proposed inception-based CNN architecture is given in Fig. 7. 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 (Szegedy et al., 2016). It provides a novel architecture to reduce the computational cost while keeping the accuracy intact. 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 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. 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 a BS. In order to efficiently train the model, the cross-entropy loss has been used coupled with adam optimizer tuned with weight decay of $1 \times 10-4$ and learning rate of $3.63 \times 10-4$. The proposed model achieved state-of-the-art accuracy of 60.55% with 20 epochs and automatic mixed-precision set to 16 floating bits.

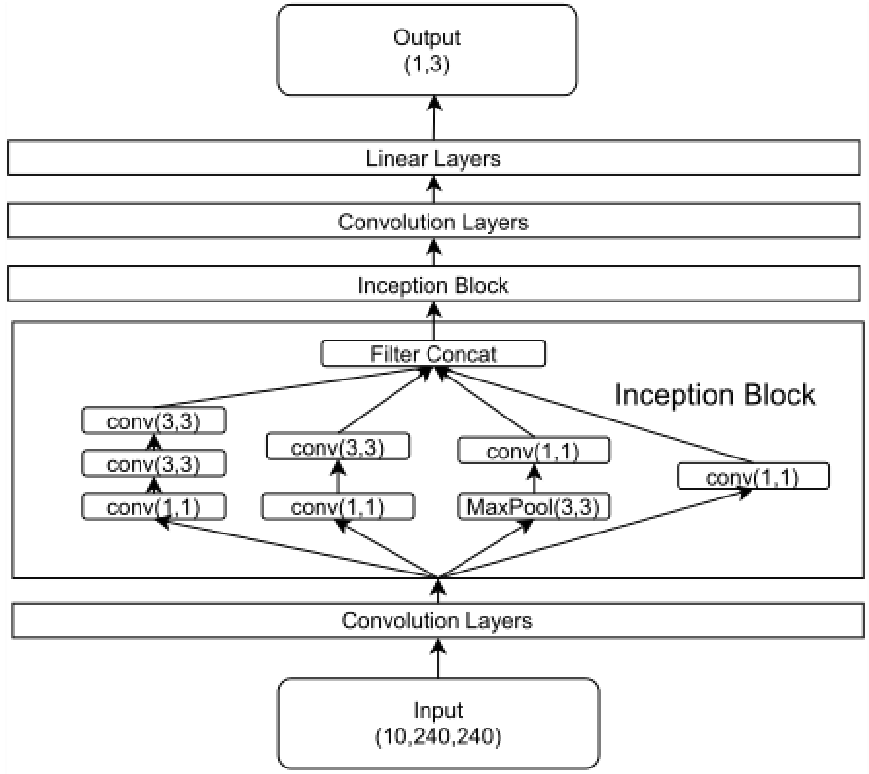

Figure 2: The proposed LiDAR inception-based CNN model

5.2 Deep Learning model using LiDAR and GPS data

The proposed CNN architecture is given in Fig. 8. The processing pipeline of LiDAR data is kept the same as discussed in section 5.1. The output size of LiDAR processing pipeline is 128. The GPS data is parallelly passed through an encoder-decoder neural network model. The output size of the GPS processing pipeline is 3. The proposed architecture is made up of a LiDAR processing pipeline and a GPS processing pipeline that have been concatenated. After concatenation, the output size of 131 is fed to FC layers. 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 a BS.

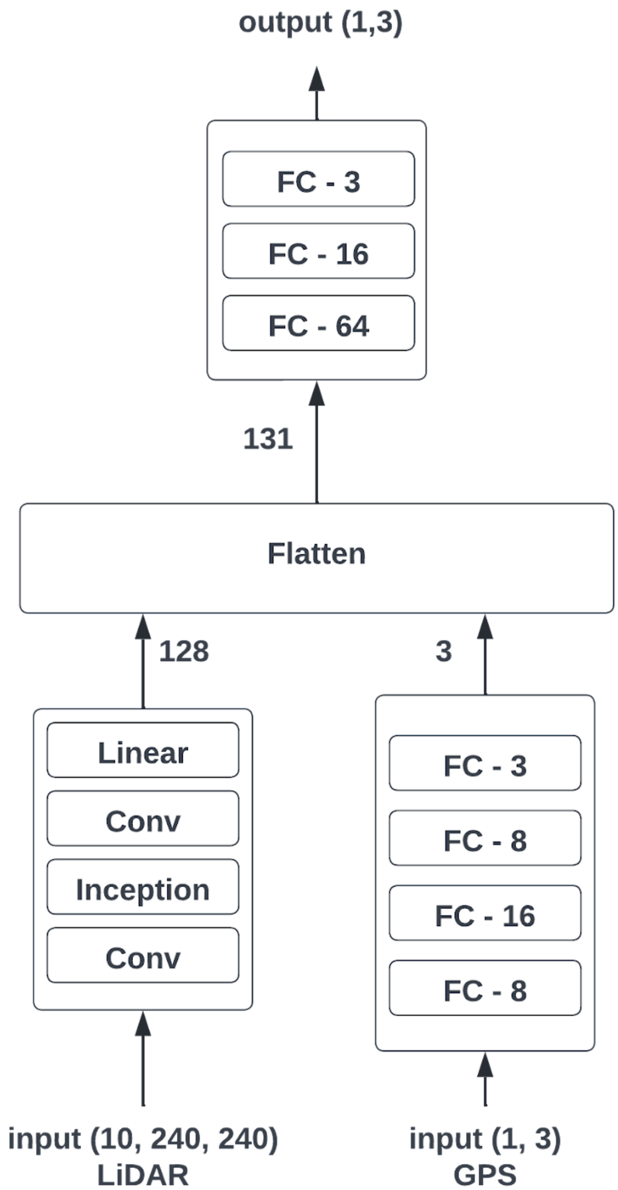

Figure 3: LiDAR and GPS CNN model

5.3 Hyperparameter tuning using particle swarm optimization on the deep learning model

PSO-based hyper-parameter tuning for getting the best value of multiple parameters used for training the model. Hybridization of parameter tuning with PSO helps for the convergence of parameter values of the proposed CNN model, and the proposed BS classification model is applied to the dataset. However, employing the backpropagation algorithm alone has a number of drawbacks. The backpropagation approach, for example, deterministically happens in local optima, making it difficult to obtain global optima, especially when a wide search space is required for the optimal solution. Also, working with the LiDAR 3D point cloud and CNN models for feature selection and classification, the number of parameters in the network is comparatively high and requires an automatic technique for the optimization of hyper-parameters. The swarm size is taken 100 particles. For the internals of PSO, we adopted a uniform parametrization throughout the experiment, where $\omega = \phi_p = \phi_g = 0.5$ are the search parameters. The termination criteria are defined as $G_{max} = 100$ is the maximum number of generations, $\delta = 10^{-4}$ is the minimum change of the best particle position, $\varepsilon = 10^{-4}$ and is minimum fitness improvement. The CNN training also includes an early termination condition, which ensures that if the accuracy does not improve after 5 training epochs, the training will be stopped. The objective function is to increase the accuracy on different set of hyperparameters such as convolutional no. of filters (*n*), convolutional filter size (*sf*), max pooling filter size (*sp*), max pooling layers stride (*l*). For [*n, sf, sp, l*], the obtained lower limit and upper limit of generation are shown table I.

Parameters	Lower limit	Upper limit
convolutional no. of filters (*n*)	1	6
convolutional filter size (*sf*)	2	6
max pooling filter size (*sp*)	2	6
max pooling layer stride (*l*)	2	6

Table I: PSO-controlled inception parameters

5.4 Federated learning on the deep learning model

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. A specific vehicle's trajectory, speed, and other features are unique to it; its local dataset will not capture all of the scenarios in the coverage region, and a NN trained on a local dataset will be erroneous and biased. Let $\boldsymbol{\theta}$ be the weights of the model used in training and $\boldsymbol{V}$ be the overall vehicles present. Therefore, $\boldsymbol{\theta}_v^i$ 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 BSs and synchronizing them. Algorithm 1 represents the training loop for FL for N communication rounds between the BS and vehicles.

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*
  end
  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 $\boldsymbol{\theta}_{\boldsymbol{v}}^{(\boldsymbol{i})} = \boldsymbol{\theta}^{(\boldsymbol{i})}, \forall v \in \boldsymbol{V}$
end

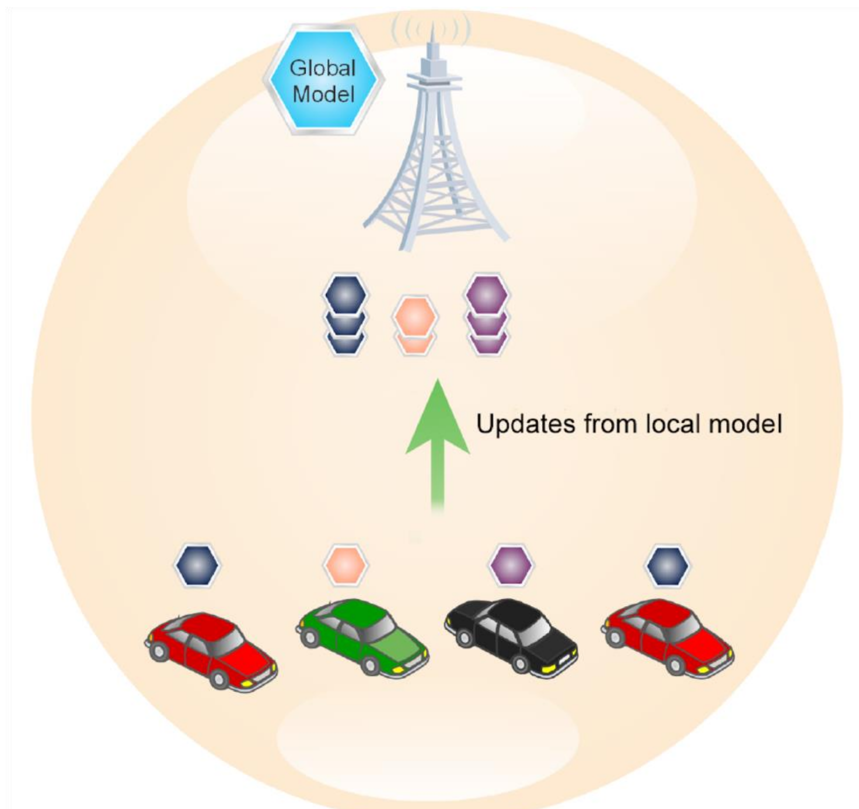

Figure 4: Process of FL

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. 9. The updated global model is then used to train the local devices for the next round. The results are computed using the mean aggregation method for aggregating the updates (Mashhadi et al., 2021). To train the CNN classifier, the cross-entropy loss (7) is calculated with an adam optimizer with an initial learning rate of 103 and a batch size of 64. The models are trained for two epochs per data set, with overall communication rounds being 10.

CHAPTER 6

Results and Discussions

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. 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.

Further, a compact DNN model with overall 339 parameters and three hidden layers of sizes 8, 16, and 8 is considered. The activation function used here is the parametric rectified linear unit with a negative slope set to 0.25, as given below in (6):

$$\begin{aligned} f(yi) &= yi, if\ yi \geq 0 \\ f(y_i) &= a_i y_i, \text{ if } y_i \leq 0 \end{aligned} \tag{6}$$

For efficient training of the model, the cross-entropy loss has been used, shown in (7), 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.

$$L_{ce} = -\sum_{i=1}^{n} t_i \log(p_i) \tag{7}$$

where n is the overall classes, here 3 or 5; 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 100m range using GPS data in DNN model.

The realistic dataset is simulated using RT and LiDAR sensor data to suggest the best BS by using CNN. The dataset contains 20,000 samples that are separated into train test validation in an 8:1:1 ratio. The 3 layer CNN gave an accuracy of 61.95% for BS selection with 2.1M parameters. For such huge parameters, memory and computation power required is enormous. Furthermore, GoogLeNet, ResNet, Inception architectures using TF were analyzed for reduction in parameters. The accuracy achieved for BS selection and their trainable parameters are shown in Table II. It can be concluded from the obtained parameters that inception-based CNN architecture was having minimum number of trainable parameters. The trainable parameters in a GoogLeNet-based CNN architecture are 132k, but the trainable parameters in an inception-based CNN architecture are 28k, which is approximately 25% of the GoogLeNet-based CNN architecture's trainable parameters with almost same accuracy. The proposed inception-based CNN model achieved an accuracy of 60.55% using LiDAR data for BS selection.

Architecture	Accuracy (%)	Trainable Parameters	Non-trainable parameters	Total parameters
3 layer CNN	61.95	2.1 M	0	2.1 M
ResNet (TF)	55.625	66 k	11.2 M	11.3 M
GoogleNet (TF)	58.125	132 k	5.6 M	5.7 M
Inception	60.55	28 k	0	28 k
PSO-Inception	63.48	18.8 k	0	18.8k

Table II: Different CNN architectures

To further improve, hyperparameter tuning is performed using PSO in the proposed inception-based CNN model. Hyperparameter tuning with PSO helps for the convergence of parameter values of the proposed CNN model, and the proposed BS classification model is applied to the dataset. While hyperparameter optimization, the best pair of hyperparameters is obtained as [1, 2, 6, 4] with the accuracy of 63.48% on the inception model using LiDAR data for BS selection. It further reduces the number of parameters from 28k to 18k as shown in Table II.

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, resulting in a 96.38% reduction in 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 shown in Table III.

Model used	Accuracy (%)
Shortest distance	27.35
GPS	35.54
LiDAR	60.55
LiDAR-GPS	63.47
LiDAR-FL	58.51

Table III: Different approaches with performance

Relationship between the number of samples and accuracy: The number of samples for training increases from 12,000 to 24,000 and 36,000, respectively. Table IV highlights the accuracy achieved on each dataset for the same epochs. The accuracy increases by increasing the number of training samples.

Samples	Accuracy (%)
12k	60.55
24k	64.45
36k	66.25

Table IV: Variation of accuracy with change in number of samples

Furthermore, to analyze and test the robustness of the model, the following experiments have been conducted. We analyze the model for 5 BSs using 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 V summarizes all the different approaches and their respective accuracies.

Training methodology	Accuracy (%)
Freezing convolution layers and fine-tuning	47.6
Fine-tuning the entire model	46.77

Table V: Comparison between different approaches for increase in BSs from 3 to 5

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 VI reports the accuracy achieved by using both transfer learning on a pre-trained model and training from scratch using the existing model.

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 pretrained weights	37.92

Table VI: Comparison between different approaches for change in location to downtown Chicago

Chapter 7

Conclusion and Future work

A methodology is proposed to realistically and accurately simulate the data for BS selection. A scheme for BS selection is proposed that leverages LiDAR data in inception-based CNN to reduce the BS search overhead and achieve greater accuracy. Further, hyperparameter tuning using PSO, a meta-heuristic technique is performed which further improves the accuracy of the model as the parameters are being optimized. 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 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 future work. Also, the system will be updated to handle the handover of the vehicle signal from one BS to another BS. The data preprocessing could be improved such that it can handle the sparse data generated from the LiDAR point cloud.

REFERENCES

1. Abuelenin, S. M., & Abul-Magd, A. Y. (2014, November). Empirical study of traffic velocity distribution and its effect on VANETs connectivity. *2014 International Conference on Connected Vehicles and Expo (ICCVE)*. http://dx.doi.org/10.1109/iccve.2014.7297577
2. Anonymous. (2013). 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*. https://doi.org/10.29012/jpc.v4i2.623
3. Gonzalez-Prelcic, N., Ali, A., Va, V., & Heath, R. W. (2017). Millimeter-Wave communication with out-of-band information. *IEEE Communications Magazine*, *55*(12), 140–146. https://doi.org/10.1109/mcom.2017.1700207
4. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8
5. Hua, Qiaozhi , Yu, Keping, & Zheng. (2019). A novel base-station selection strategy for cellular vehicle-to-everything (C-V2X) communications. *Applied Sciences*, *9*(3), 0–556.
6. *Index of /*. (n.d.). Retrieved May 22, 2022, from https://planet.osm.org
7. Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K.-C., & Hanzo, L. (2017). Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, *24*(2), 98–105. https://doi.org/10.1109/mwc.2016.1500356wc
8. Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*. Retrieved May 23, 2022, from http://dx.doi.org/10.1109/icnn.1995.488968
9. Klaine, P. V., Imran, M. A., Onireti, O., & Souza, R. D. (2017). A survey of machine learning techniques applied to self-organizing

cellular networks. *IEEE Communications Surveys & Tutorials*, *19*(4), 2392–2431. https://doi.org/10.1109/comst.2017.2727878

10. Klautau, A., Batista, P., Gonzalez-Prelcic, N., Wang, Y., & Heath, R. W. (2018, February). 5G MIMO data for machine learning: Application to beam-selection using deep learning. *2018 Information Theory and Applications Workshop (ITA)*. http://dx.doi.org/10.1109/ita.2018.8503086
11. Klautau, A., Gonzalez-Prelcic, N., & Heath, R. W. (2019). LIDAR Data for Deep Learning-Based mmWave Beam-Selection. *IEEE Wireless Communications Letters*, *8*(3), 909–912. https://doi.org/10.1109/lwc.2019.2899571
12. Krajzewicz, D. (2010). Traffic simulation with SUMO – simulation of urban mobility. In *International Series in Operations Research & Management Science* (pp. 269–293). Springer New York. http://dx.doi.org/10.1007/978-1-4419-6142-6_7
13. Lili Wei, Hu, R., Yi Qian, & Geng Wu. (2014). Key elements to enable millimeter wave communications for 5G wireless systems. *IEEE Wireless Communications*, *21*(6), 136–143. https://doi.org/10.1109/mwc.2014.7000981
14. Litman, T. (2017). Autonomous vehicle implementation predictions.
15. Lorenzo, P. R., Nalepa, J., Ramos, L. S., & Pastor, J. R. (2017, July 15). Hyper-parameter selection in deep neural networks using parallel particle swarm optimization. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. http://dx.doi.org/10.1145/3067695.3084211
16. Marathe, N. V., Desai, U. B., & Merchant, S. N. (2008). Base station selection strategy in multihop cellular networks: A new approach. *2008 International Conference on Signal Processing, Communications and Networking*. http://dx.doi.org/10.1109/icscn.2008.4447227
17. Mashhadi, M. B., Jankowski, M., Tung, T.-Y., Kobus, S., & Gunduz, D. (2021). Federated mmWave Beam Selection Utilizing LIDAR Data.

IEEE Wireless Communications Letters, *10*(10), 2269–2273. https://doi.org/10.1109/lwc.2021.3099136

18. Mishra, B., Garg, D., Narang, P., & Mishra, V. (2020). A hybrid approach for search and rescue using 3DCNN and PSO. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05001-7
19. Mishra, V. K., & Sengupta, A. (2014). MO-PSE: Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design. *Advances in Engineering Software*, *67*, 111–124. https://doi.org/10.1016/j.advengsoft.2013.09.001
20. Nguyen, H., Tran, & Seon, C. S. (2019). Federated learning over wireless networks: Optimization model design and analysis. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 1387–1395.
21. Qin, Z., Li, G. Y., & Ye, H. (2021). Federated learning and wireless communications. *IEEE Wireless Communications*, *28*(5), 134–140. https://doi.org/10.1109/mwc.011.2000501
22. Schaubach, K. R., Davis, N. J., & Rappaport, T. S. (n.d.). A ray tracing method for predicting path loss and delay spread in microcellular environments. *[1992 Proceedings] Vehicular Technology Society 42nd VTS Conference - Frontiers of Technology*. Retrieved May 23, 2022, from http://dx.doi.org/10.1109/vetec.1992.245274
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016, June). Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. http://dx.doi.org/10.1109/cvpr.2016.308
24. Tran, N. H., Bao, W., Zomaya, A., Nguyen, M. N. H., & Hong, C. S. (2019, April). Federated learning over wireless networks: Optimization model design and analysis. *IEEE INFOCOM 2019 - IEEE Conference*

on Computer Communications. http://dx.doi.org/10.1109/infocom.2019.8737464

25. *Webots: Robot simulator*. (n.d.). Retrieved May 22, 2022, from http://www.cyberbotics.com
26. Yangli-ao Geng, Ming Liu, Qingyong Li, & Ruisi He. (2019). Introduction of machine learning. In *Applications of Machine Learning in Wireless Communications* (pp. 1–65). Institution of Engineering and Technology. http://dx.doi.org/10.1049/pbte081e_ch1
27. Yun, Z., & Iskander, M. F. (2015). Ray Tracing for radio propagation modeling: Principles and applications. *IEEE Access, 3*, 1089–1100. https://doi.org/10.1109/access.2015.2453991