IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 September 2022, accepted 5 October 2022, date of publication 1 November 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218622

RESEARCH ARTICLE

Machine Learning Approaches for Radio Propagation Modeling in Urban Vehicular Channels

KHALIL AHMAD AND SAJJAD HUSSAIN, (Member, IEEE)
School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
Corresponding author: Sajjad Hussain (sajjad.hussain2@seecs.edu.pk)

ABSTRACT The use of vehicular communications is anticipated to improve safety in road traffic. The traditional radio channel models that describe the effects of radio wave propagation in dynamic vehicular environments have their own limitations. In this paper, machine learning (ML) techniques are applied for radio channel modeling in urban vehicular environments. A large data set of path loss (PL) and root-mean-square Delay spread (RMS-DS) is computed using ray-tracing for a Line-of-Sight (LOS) straight road and a Non-Line-of-Sight (NLOS) intersection road scenario. Fourteen input features are used to train three ML models for vehicular channel prediction. The models considered in this work include Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Random Forest (RF). The results show that RF gives better performance than MLP and CNN models in the prediction of PL and RMS-DS in urban vehicular channels.

INDEX TERMS Vehicular channel modeling, machine learning, ray tracing, multi-layer perceptron, convolutional neural network, random forest.

I. INTRODUCTION

The idea of vehicular communication is to provide a secure means of exchanging information among vehicles. The key challenges in today's vehicular communication systems are assuring safety, efficiency, and traffic flow while maintaining a comfortable journey. These challenges can be addressed by establishing reliable vehicular communication. This advocates the development of accurate radio channel models for vehicular communication. Vehicular channels differ significantly from cellular communication channels due to the constantly changing surroundings in which they operate. Traditional approaches for vehicular channel modeling include measurement based empirical models [1], [2], [3], [4], non-geometry based stochastic models [5], [6], [7], geometry based stochastic models [8], [9], [10], [11], and ray-based deterministic models [12], [13], [14].

Machine learning (ML) methods have recently been applied to radio channel modeling [15]. These techniques can quickly adapt to non-linearity in radio channel parameters due to changing environments that make these models well suited for real-time channel modeling in vehicular scenarios. Several ML models including Support Vector Machines (SVM), Artificial Neural Networks (ANN), Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN) and Random Forest (RF) have recently been proposed to predict radio channel parameters in radio networks [16], [17]. Huang et al. in [18] investigated SVM, RF and ANN models for identification of Line-of-Sight (LOS) in vehicle-to-vehicle (V2V) channels at 5.9 GHz. Time-varying angular properties of the channel including rising time, Rician K-factor, root-mean-square Delay spread (RMS-DS), and kurtosis and skewness of received power are used for training of ML models. Yang et al. in [19] used an SVM-based model for the prediction of Angle-of-Arrival (AOA) in vehicular channels. Real-time measured data of power delay profile (PDP) and path loss (PL) are used to train the suggested model. Reference [20] reported a Back Propagation Neural Network (BPNN) based model to identify the vehicular channel scenarios in urban, highway and tunnels. Ramya et al. in [21] used RF model to predict PL in V2V channels.

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

The performance of the proposed model was assessed for different combinations of input features. It was observed that prediction accuracy improved with additional input features. Turan et al. in [22] also developed an RF model to predict PL in vehicular visible light communications (VVLC) and IEEE 802.11p channels using real world measured data. The proposed model performed well as compared to curve fitting models. In another work by Turan et al. in [23] MLP, RF and Radial Basis Function Neural Network (RBF-NN) for PL and channel frequency response prediction in VVLC channels are proposed.

Although the ML based PL prediction models have been extensively studied in recent years, most of the work in literature has been reported for indoor and outdoor cellular networks. A little work has been reported for ML based PL prediction in vehicular networks. This paper addresses this gap and investigates three ML approaches namely MLP, CNN and RF to predict PL and RMS-DS in both LOS and NLOS urban vehicular channels.

The rest of this paper is organized as follows. Section II discusses the development of ML models used in this paper. Data set preparation, list of input features, and a brief overview of MLP, CNN and RF is provided. Optimization of the proposed models is also explained. PL and RMS-DS prediction results are discussed in section III. Finally, conclusion is provided in section IV.

II. DEVELOPMENT OF MACHINE LEARNING MODELS

The development of ML models for vehicular channel modeling is discussed in this section. A large data set of radio channel parameters is generated using ray-tracing. The data set is pre-processed and divided into a train data set and a test data set. The proposed ML models are trained using the train data set and optimized for model selection. Block diagram of proposed methodology is shown in Fig. 1. Further details are provided in the following subsections.

A. DATA SET AND LEARNING PARAMETERS

The first step in supervised ML models is to obtain a large data set that has pre-computed the value of the target output variable for different possible combinations of all the input variables. These input variables are also called input features in the context of ML. The details of data set preparation and input features are given in the following.

1) DATA SET GENERATION

In this paper, the data set is generated using an in-house ray-tracing simulation tool [24], [25], [26], [27]. Two most common V2V communication scenarios including LOS on a straight road and NLOS on a road intersection in an urban setting are simulated. The scenarios simulated in this paper are drawn from the Munich city [28] and are illustrated in Fig. 2. The vehicles are either moving towards each other (LOS case) or towards an urban street intersection (NLOS case). The height of both the transmitter (Tx) and receiver (Rx) antenna is assumed 1.6m. Multi-path channel with up to four wall reflections and one corner diffraction is simulated at carrier frequency of 5.9 GHz. Table 1 shows the simulation parameters of the ray-tracing tool to generate the data set used in this paper. Two output parameters including PL and RMS-DS are computed at each time instance. PL is the ratio of the power radiated by Tx antenna to the power received by the Rx antenna and is given as follows.

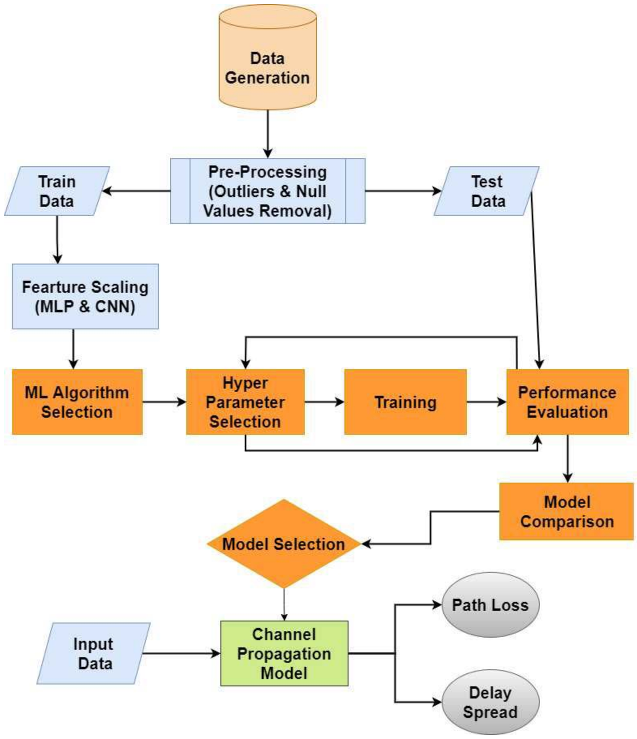

FIGURE 1. Block diagram of the proposed framework for vehicular channel modeling using ML.

$$P_{loss}(dB) = 10log\frac{P_t}{P_r} = P_t(dB) - P_r(dB) \quad (1)$$

where P_r and P_t are the received and transmitted power respectively. Total received power is computed using superposition of electric fields of all the valid rays arriving at the RX.

RMS-DS is the square root of the second central moment of PDP and is given as follows [14].

$$\tau_{rms} = \sqrt{\frac{\sum_{i=1}^{n} P_i(\tau_i - \overline{\tau})^2}{\sum_{i=1}^{n} P_i}} \quad (2)$$

where P_i, and τ_i represent power and excess delay of the i^{th} ray respectively, and n is the total number of rays. $\overline{\tau}$ represents mean delay and is given by (3).

$$\overline{\tau} = \frac{\sum_{i=1}^{n} P_i\tau_i}{\sum_{i=1}^{n} P_i} \quad (3)$$

A total of 24, 500 data samples that include 14, 000 data samples for LOS scenario and 10, 500 data samples for NLOS scenario are computed to produce the data set used in this paper. Each sample is computed using the combination of

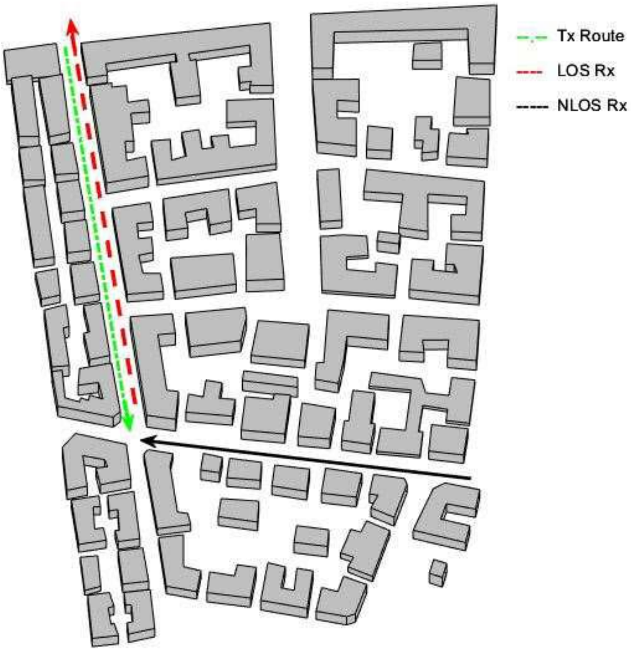

FIGURE 2. Simulation environment with a LOS and NLOS scenario used for data set generation.

TABLE 1. The simulation parameters of ray-tracing tool to generate the data set.

Parameter	Chosen value
Wall reflections	Up to 4
Corner Diffraction	1
Tx power	14 dB
Frequency	5.9 GHz
Antenna type	Omni-directional
Building Material	Concrete
Wall Permittivity (ϵ_r)	3.75
wall conductivity (σ)	0.137

fourteen different input features. A standard Intel® Core™ i5 computer with 16 GB RAM is used in simulations.

2) LIST OF INPUT FEATURES

The input features for the proposed models are Tx coordinates, Rx coordinates, separation between Tx and Rx, number of buildings penetrated by direct line joining Tx and Rx, distance covered inside buildings by direct line joining Tx and Rx, distance covered outside buildings by direct line joining Tx and Rx, width of street in which Tx is located, width of street in which Rx is located, distance of Tx from side corner, distance of Rx from side corner, LOS or NLOS case, and speed of the vehicles. These input features are listed in Table 2. Figure 3 shows graphical representation of some of the features for an NLOS example. The total indoor and outdoor distance is given by

$$d_{in} = \sum_{i} d_{in}^{i} \tag{4}$$

$$d_{out} = \sum_{i} d_{out}^{i} \tag{5}$$

TABLE 2. The detailed description of input features.

Feature	Notation	Description	Min value	Max value
Tx Coordinate	(x_t, y_t)	coordinates of the Tx.	(0,0)	(350,450)
Rx Coordinate	(x_r, y_r)	coordinates of the Rx.	(0,0)	(350,450)
Separation	r	Distance between Tx and Rx.	8	235
Number of shadowed buildings	n	Number of buildings penetrated by direct line joining Tx and Rx.	0	8
Indoor distance	d_{in}	Distance covered inside buildings by direct line joining Tx and Rx.	0	156
Outdoor distance	d_{out}	Distance covered outside buildings by direct line joining Tx and Rx.	107	235
Tx street width	w_t	Tx street width.	19	29
Rx street width	w_r	Rx street width.	21	31
Tx from edge	Td_E	Distance of Tx from side corner	5	11
Rx from edge	Rd_E	Distance of Rx from side corner.	6	13
Link type	L/N	LOS or NLOS scenario	0 for NLOS	1 for LOS
Speed of vehicles	V	Speed of Tx and Rx vehicles .	36	50

In example given in Fig. 3 two buildings are penetrated by the direct line between Tx and Rx, so n is 2. It can be noted that separation r is the sum of indoor and outdoor distances as follows.

$$r = d_{in} + d_{out} \tag{6}$$

3) DATA PRE-PROCESSING

The first step in pre-processing of data set includes removal of outliers and null values. Quantile ranges method for outlier removal and *Pandas* library was used to remove null values in the data set. *Pandas* is a well known *Python* library used for the analysis and manipulation of input data. Scaling of input features is performed in the next step. The range of values of the input features varies by a large extent e.g. the separation between Tx and Rx ranges from 8m to 235m whereas the number of buildings penetrated by direct line joining Tx and Rx ranges from 0 to 8 and so on. Feature scaling is performed before training the models to make sure that the values of all the features vary in the same range so that the cost function associated with ML models can easily converge for better prediction accuracy. In this paper, Min-Max normalization is applied for feature scaling as follows.

$$\overline{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

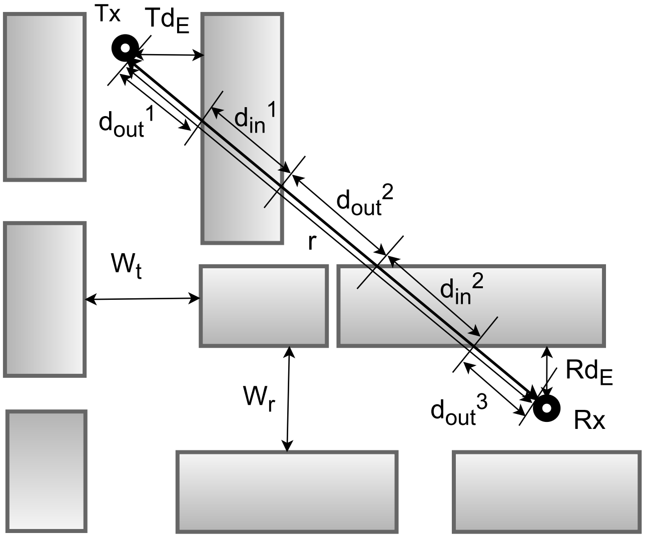

FIGURE 3. Visual representation of input features used in ML models.

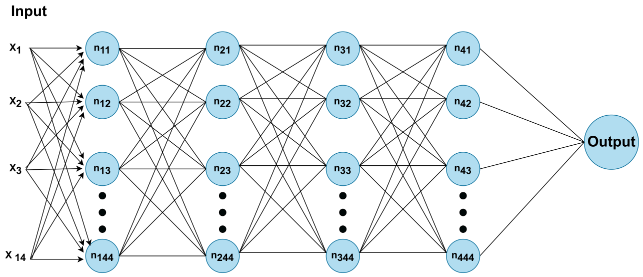

FIGURE 4. Block diagram of MLP architecture with each hidden layer consisting of 44 neurons in the proposed model.

where x_{min} and x_{max} represents the minimum and maximum value of the input feature respectively, x represents the actual value and $\overline{x}$ represents the normalized value. After removing of outliers and null values, and performing data normalization, the data set is divided into training data set and testing data set that constitutes 80% and 20% of total samples respectively.

B. PROPOSED MACHINE LEARNING MODELS

Two well-known deep learning models namely MLP and CNN, and an ensemble learning model RF are developed in this work. A brief overview of each of the models is provided in the following.

1) MULTI-LAYER PERCEPTRON

The MLP is a widely used artificial neural network for solving a number of ML problems [29]. MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer consists of a number of nodes that are called neurons. A non-linear activation function exists for each node in MLP, allowing it to solve complex learning tasks. A generic block diagram of MLP model is depicted in Fig.4.

MLP performs the training process in two steps. Firstly, in forward propagation step, network weights are computed and output is predicted by propagating input data across the network from input layer to output layer. The output $\hat{y}$ of the MLP model for some input data 'x_i' is obtained as follows (8).

$$\hat{y} = \sigma\Big(\sum_{i} w_i * x_i + b \Big) \tag{8}$$

where w_i is the associated weight matrix with input x_i, b is the bias and σ represents the activation function.

Secondly, in the back-propagation phase, MLP output is compared with the actual output to compute the error signal. The error is propagated from output layer to input layer through the network, and the weights and biases are updated accordingly. This method is repeated until a predefined criteria, such as the mean square error being almost zero, is achieved.

2) CONVOLUTIONAL NEURAL NETWORK

CNNs are well-known deep learning models that can find patterns in data without requiring manual feature extraction and can be retrained for new recognition tasks. CNNs are ideal models to deal with ML problems having massive data sets such as images. A comprehensive CNN architecture is shown in Fig. 5. It consists of three different types of layers [30].

1) Convolutional Layers: Convolutional layers extract features from input data by convolving the input data with the kernels. The convolution of input data vector 'X' with a kernel 'K' is obtained by (9).

$$z_j = X * K = \sum_{i=-s}^{s} x_{j-i} k_i \tag{9}$$

here z_j, X, and K are output, an input data with length 'L', and filter or kernel respectively while the * shows convolution operator. After the convolution layer, nonlinearity is applied by a non-linear function to generate or limit the output of the node.

2) Pooling Layers: The pooling layer's objective is to reduce the computational cost. The pooling layer reduces the parameters by performing down sampling.

3) Fully connected layers: Fully connected layers in CNN are like the layers found in traditional ANNs. The nodes in the fully connected layers are connected to all the nodes in both previous and next layers. These layers are applied at the end of the architecture that work as classifiers.

3) RANDOM FOREST

RF model has been proven a successful classifier and regression method for practical problems. RF is an ensemble learning based algorithm in which multiple decision trees are used for prediction. RF dramatically reduces the over-fitting problem through training each decision tree on different random subsets. RF model consists of two segments.

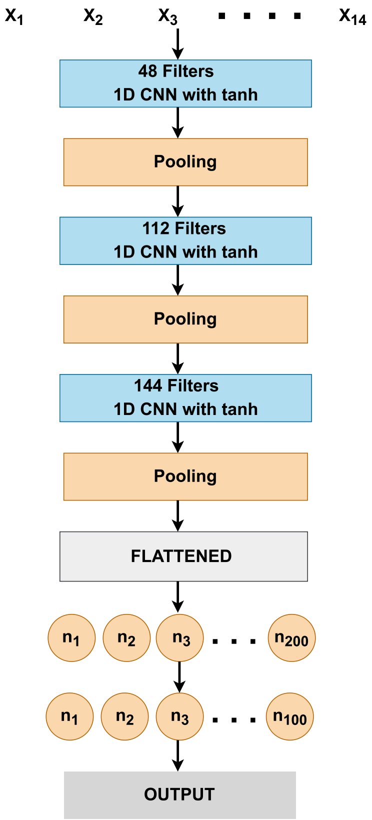

FIGURE 5. Block diagram of CNN architecture with 9 layers.

1) Boot-strapping: Boot-strapping is the process of creating a mini data set by randomly selecting samples from the larger data set. The sub-sample size is chosen to be less than the original data size and samples can be included more than once as these are picked randomly from the original sample space [31].
2) Tree-making: In this phase, a decision tree is made up of 'k' number of features randomly selected from a subset, which are less than the total number of samples.

These two steps are repeated a certain number of times and the trees thus formed are combined to make a RF. For a regression problem, the output of all the trees is combined and the average value is taken as the predicted value. Figure 6 shows a generic block diagram of RF. The result of RF model for a regression problem with N number of trees is represented by (10)

$$\hat{y} = \frac{1}{N}(\sum_{i=1}^{N} y_i) \tag{10}$$

here $\hat{y}$ is the predicted value and y_i is the output of each decision tree.

C. TRAINING AND OPTIMIZATION OF MACHINE LEARNING MODELS

The ML models proposed in this work are trained using *Python* on *Google Colab* environment [32]. The mean

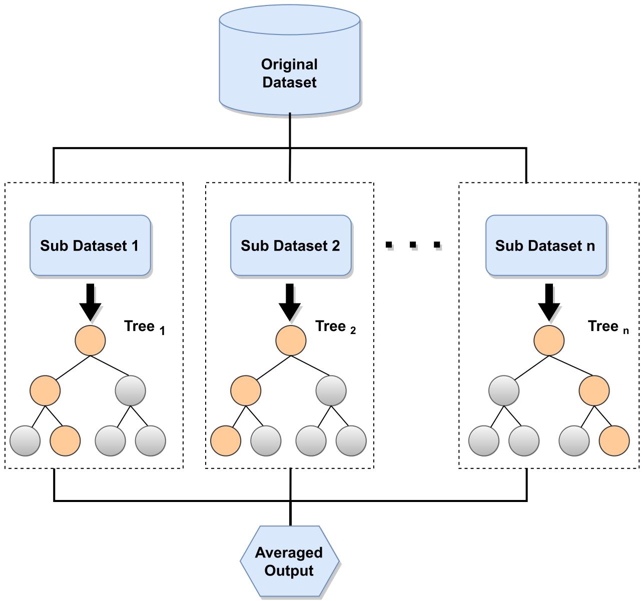

FIGURE 6. Block diagram of RF model.

absolute error (MAE), root mean square error (RMSE), and mean square error (MSE) scores of the suggested models are assessed to determine the model performance. Equations (11), (12), and (13) are used to calculate MAE, MSE, and RMSE, respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{11}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{12}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \tag{13}$$

where $\hat{y}$ and y are the predicted and computed values of the output parameter respectively, N denotes the total number of samples, and i is the index of each sample.

1) HYPER-PARAMETERS TUNING

The most challenging task is to determine the optimal hyper-parameters for the models. In this study, random search and grid search are used to obtain the optimized parameters. In random search tuning process, parameters are chosen at random from a set of hyper-parameters. In grid search, all of the parameters are evaluated in order to choose the best ones. Deep neural networks consist of a large number of hyper-parameters, and conducting a grid search operation for each would take considerable time to train. As a result, random search is applied to find the best MLP and CNN parameters. Grid search is used to tune the hyper parameters of the RF.

TABLE 3. The design parameters used for MLP Model.

Layer No.	1	2	3	4	5 (output)
Neurons	44	44	44	44	1
Activation Function	tanh	tanh	tanh	tanh	linear

TABLE 4. The Parameters used for CNN Model, having a total of 9 layers.

Layer type	Filters/ Neurons	Kernel/ pool size	Pading	Activation
Convol:1	48	5	same	tanh
Pooling:1	-	-	-	tanh
Convol:2	112	3	same	-
Pooling:2	-	-	-	-
Convol:3	144	3	same	tanh
Pooling:3	-	-	-	-
FC:1	200	-	-	tanh
FC:2	100	-	-	tanh
FC:3	1	-	-	linear

K-fold cross validation is used to further improve the model's performance. In this study, k is set to ten folds ($k = 10$) for the k-fold cross validation procedure. The whole data set is divided into 10 sub-groups. The model is trained using 9 sub-groups and tested using a remaining sub-group in the first phase. In second phase, a set of alternate 9 sub-groups is used for training and a sub-group (different from the testing sub-group during previous phase) is used for testing. This process is repeated until the model is trained and tested for all 10 sub-groups of the data set.

TensorFlow library [33] is used to develop the proposed MLP and CNN models while RF model in this work is implemented using *sklearn* library [34]. The input size for both the MLP and the CNN is set to 14 corresponding to the input features. Further details of setup of each of the models is given below.

- **MLP**: A 5 layer deep MLP model is developed with stochastic gradient descent as an optimizer. The learning rate has been selected as 0.0001 and momentum is set to 0.9. The 'Mean square error' is selected as loss function. Parameters for the MLP model are listed in Table 3.
- **CNN**: A 9 Layer deep CNN model is implemented with stochastic gradient descent as optimizer. The learning rate is set to 0.0001 and momentum is set to 0.9. The CNN is made up of 3 convolutional (Conv), 3 max-pooling (pool) and 3 fully connected (FC) layers. The mean square error is chosen as loss function for CNN. The model is trained for 200 epochs. The details of CNN parameters are shown in Table 4.
- **RF**: Please note that the RF method does not require data normalization, so it uses 80% of the non-normalized input data for training. Table 5 shows the details of optimized parameters for RF obtained using grid search.

TABLE 5. The Parameters used in designing RF Model.

Parameter	Magnitude/ Value
Estimators	100
Criterion	Gini
Max depth	None
Max features	Auto
Min samples leaf	1

III. RESULTS

Three ML models discussed above are optimized to predict PL and RMS-DS in urban vehicular channels. The detailed discussion on results comparison for output parameters is given in the following.

A. PATH LOSS PREDICTION

The trained ML models including MLP, CNN and RF are used to predict PL values for each sample in test data set. The performance of the ML models is assessed separately for LOS and NLOS scenarios. Table 6 summarizes RMSE values for the three models in straight road and road intersection scenarios. It can be observed that RMSE values for MLP, CNN and RF models for LOS scenario are 2.19 dB, 2.03 dB and 0.63 dB respectively. Likewise, the RMSE values for MLP, CNN and RF models in NLOS scenario are 4.06 dB, 2.64 dB and 0.44 dB respectively. The RF model gives the best PL prediction results with RMSE values less than 1 dB in both LOS and NLOS scenarios. It must also be noted that CNN and MLP models also performed well with reasonable accuracy. The actual and predicted PL values using RF model for straight road scenario and road intersection scenario are given in Figs. 7 and 8, respectively. The horizontal axis is the distance in meters between Tx and Rx.

TABLE 6. RMSE comparison of all models for PL predictions in LOS and NLOS scenarios.

Model	LOS case (dB)	NLOS case (dB)
MLP	2.19	4.06
CNN	2.03	2.64
RF	0.63	0.44

The RMSE plot for PL prediction using MLP, CNN and RF models in LOS and NLOS scenarios are shown in Figs. 9 and 10 respectively. The RMSE plot for MLP and CNN is generated as a function of number of epochs for MLP and CNN models, and for RF the plot is generated as a function of number of estimators. It was observed that the execution time for training the RF model on Google Co-lab GPUs is 18 times less than required for MLP model and 25 times less than required for CNN model for 200 epochs.

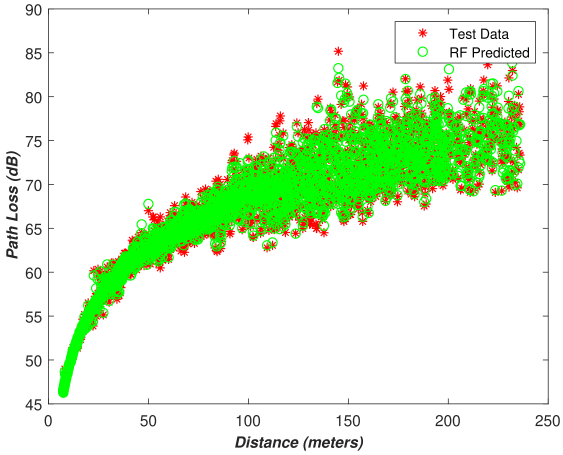

FIGURE 7. Actual and predicted values of PL using RF model for LOS scenario.

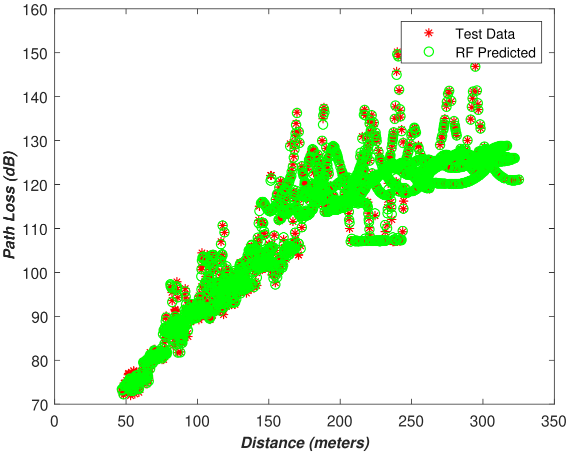

FIGURE 8. Actual and predicted values of PL using RF model for NLOS scenario.

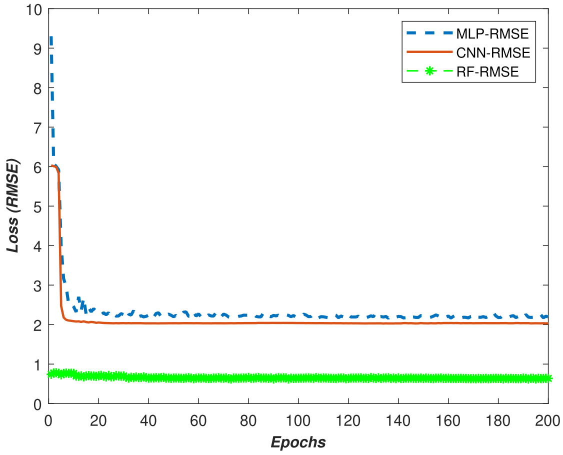

FIGURE 9. Comparison of RMSE for PL prediction for all three ML models under LOS scenario.

B. RMS DELAY SPREAD PREDICTION

The same trained ML models including MLP, CNN and RF are used to predict RMS-DS values for each sample in test data set. The performance of the ML models is assessed separately for LOS and NLOS scenarios. RF model again gives better prediction accuracy than both the MLP and CNN models. Table 7 summarises RMSE performance of proposed ML models in both LOS and NLOS scenario. It can be observed that RMSE values of MLP, CNN and RF models for straight road scenario are 10.6ns, 8.30ns and 5.88ns respectively. Likewise, the RMSE values of MLP, CNN and RF models for road intersection scenario are 11.0ns, 8.53ns and 1.8ns respectively. The actual and predicted RMS-DS values using RF model for LOS and NLOS scenario are given in

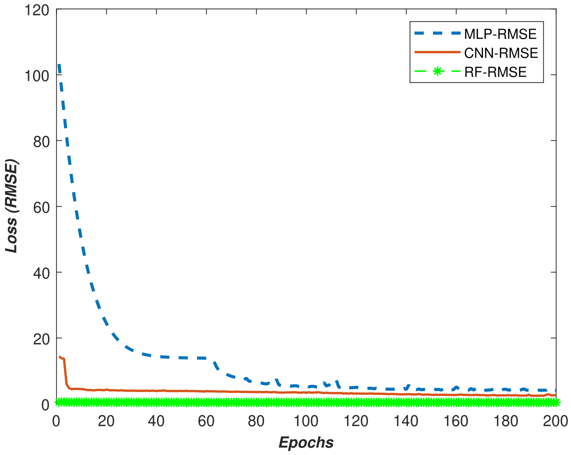

FIGURE 10. Comparison of RMSE for PL prediction for all three ML models under NLOS scenario.

TABLE 7. RMSE comparison of all models for RMS-DS predictions in LOS and NLOS scenarios.

Model	LOS case (ns)	NLOS case (ns)
MLP	10.6	11.0
CNN	8.30	8.53
RF	5.88	1.80

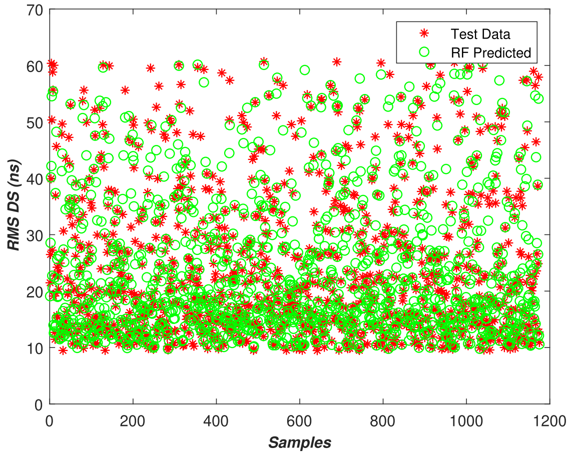

FIGURE 11. Actual and predicted values of RMS-DS using RF model for LOS scenario.

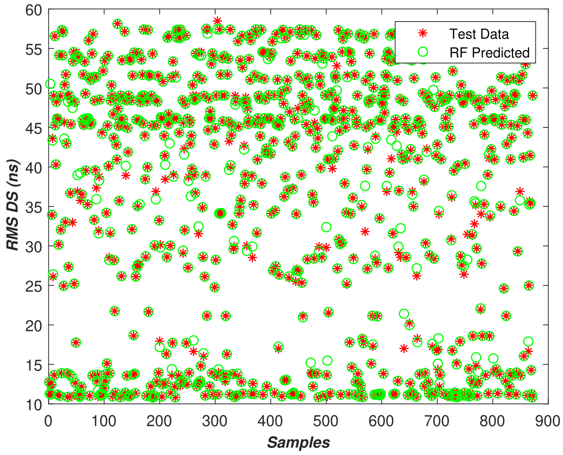

FIGURE 12. Actual and predicted values of RMS-DS using RF model for NLOS scenario.

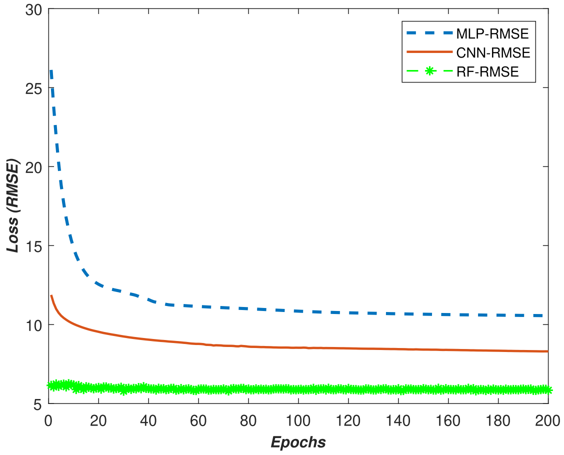

FIGURE 13. Comparison of RMSE for RMS-DS prediction for all three ML models under LOS scenario.

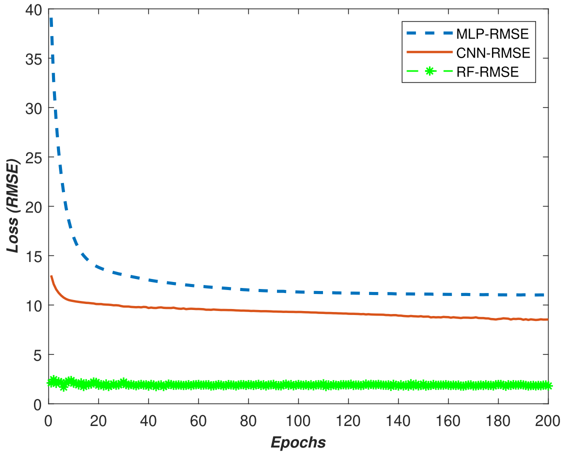

FIGURE 14. Comparison of RMSE for RMS-DS prediction for all three ML models under NLOS scenario.

Figs. 11 and 12 respectively. The convergence of RMSE for RMS-DS prediction using MLP, CNN and RF models in LOS and NLOS scenarios as a function of number of epochs are shown in Figs. 13 and 14 respectively.

IV. CONCLUSION

ML models including RF, MLP and CNN are investigated for PL and RMS-DS prediction in urban vehicular channels. The proposed models are trained using the data set generated through ray-tracing simulations. The data pre-processing, hyper-parameters optimization and K-fold cross validation for improving the model performance are included in the proposed framework. RMSE for RF model in PL prediction is 0.63 dB and 0.44 dB for LOS and NLOS scenario respectively. MLP and CNN models performed reasonably well in PL prediction with maximum RMSE values of 2.64 dB and 4.06 dB respectively. A similar performance in RMS-DS prediction is observed. RF model gives the best results with minimum RMSE value of 1.80 ns, whereas MLP model incurred maximum RMSE value of 11 ns in RMS-DS prediction. The results show that ML models can be used for radio propagation modeling in urban vehicular channels.

REFERENCES

[1] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.

[2] O. Onubogu, K. Ziri-Castro, D. Jayalath, K. Ansari, and H. Suzuki, "Empirical vehicle-to-vehicle pathloss modeling in highway, suburban and urban environments at 5.8 GHz," in *Proc. 8th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2014, pp. 1–6.

[3] M. Nilsson, C. Gustafson, T. Abbas, and F. Tufvesson, "A path loss and shadowing model for multilink vehicle-to-vehicle channels in urban intersections," *Sensors*, vol. 18, no. 12, p. 4433, Dec. 2018.

[4] S. Zelenbaba, B. Rainer, M. Hofer, D. Loschenbrand, A. Dakic, L. Bernado, and T. Zemen, "Multi-node vehicular wireless channels: Measurements, large vehicle modeling, and hardware-in-the-loop evaluation," *IEEE Access*, vol. 9, pp. 112439–112453, 2021.

[5] Q. Wang, Z. Zhong, B. Ai, and L. Liu, "A tapped delay-based channel model for vehicle-to-vehicle communications," in *Proc. 5th IET Int. Conf. Wireless, Mobile Multimedia Netw. (ICWMMN)*, 2013, pp. 132–135.

[6] Y. Li, B. Ai, D. G. Michelson, S. Lin, Q. Wang, and Z. Zhong, "A method for generating correlated taps in stochastic vehicle-to-vehicle channel models," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.

[7] N. Hassan, M. Käske, C. Schneider, G. Sommerkorn, R. Thomä, and D. Matolak, "Measurement-based determination of parameters for non-stationary TDL models with reduced number of taps," *IET Microw., Antennas Propag.*, vol. 14, no. 14, pp. 1719–1732, Nov. 2020.

[8] X. Cheng, C.-X. Wang, D. I. Laurenson, S. Salous, and A. V. Vasilakos, "An adaptive geometry-based stochastic model for non-isotropic MIMO mobile-to-mobile channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4824–4835, Sep. 2009.

[9] Y. Yuan, C.-X. Wang, Y. He, M. M. Alwakeel, and E. M. Aggoune, "3D wideband non-stationary geometry-based stochastic models for non-isotropic MIMO vehicle-to-vehicle channels," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6883–6895, Dec. 2015.

[10] C. Huang, R. Wang, P. Tang, R. He, B. Ai, Z. Zhong, C. Oestges, and A. F. Molisch, "Geometry-cluster-based stochastic MIMO model for vehicle-to-vehicle communications in street canyon scenarios," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 755–770, Feb. 2021.

[11] C. Gustafson, K. Mahler, D. Bolin, and F. Tufvesson, "The COST IRACON geometry-based stochastic channel model for vehicle-to-vehicle communication in intersections," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2365–2375, Mar. 2020.

[12] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5G and beyond wireless communications: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 10–27, 1st Quart., 2019.

[13] A. W. Mbugua, Y. Chen, L. Raschkowski, L. Thiele, S. Jaeckel, and W. Fan, ''Review on ray tracing channel simulation accuracy in sub-6 GHz outdoor deployment scenarios,'' *IEEE Open J. Antennas Propag.*, vol. 2, pp. 22–37, 2021.

[14] L. Xiong, Z. Yao, H. Miao, and B. Ai, ''Vehicle-to-vehicle channel characterization based on ray-tracing for urban road scenarios,'' *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–15, Mar. 2021.

[15] S. M. Aldossari and K.-C. Chen, ''Machine learning for wireless communication channel modeling: An overview,'' *Wireless Pers. Commun.*, vol. 106, no. 1, pp. 41–70, May 2019.

[16] C. Huang, R. He, B. Ai, A. F. Molisch, B. K. Lau, K. Haneda, B. Liu, C.-X. Wang, M. Yang, C. Oestges, and Z. Zhong, ''Artificial intelligence enabled radio propagation for communications—Part I: Channel characterization and antenna-channel optimization,'' *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3939–3954, Jun. 2022.

[17] C. Huang, R. He, B. Ai, A. F. Molisch, B. K. Lau, K. Haneda, B. Liu, C.-X. Wang, M. Yang, C. Oestges, and Z. Zhong, ''Artificial intelligence enabled radio propagation for communications—Part II: Scenario identification and channel modeling,'' *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3955–3969, Jun. 2022.

[18] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang, B. Ai, and Z. Zhong, ''Machine learning-enabled LOS/NLOS identification for MIMO systems in dynamic environments,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3643–3657, Jun. 2020.

[19] M. Yang, B. Ai, R. He, C. Huang, Z. Ma, Z. Zhong, and J. Li, ''Machine-learning-based fast angle-of-arrival recognition for vehicular communications,'' *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1592–1605, Feb. 2021.

[20] M. Yang, B. Ai, R. He, C. Shen, M. Wen, C. Huang, J. Li, Z. Ma, L. Chen, X. Li, and Z. Zhong, ''Machine-learning-based scenario identification using channel characteristics in intelligent vehicular communications,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3961–3974, Jul. 2021.

[21] P. M. Ramya, M. Boban, C. Zhou, and S. Stanczak, ''Using learning methods for V2V path loss prediction,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[22] B. Turan, A. Uyrus, O. N. Koc, E. Kar, and S. Coleri, ''Machine learning aided path loss estimator and jammer detector for heterogeneous vehicular networks,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[23] B. Turan and S. Coleri, ''Machine learning based channel modeling for vehicular visible light communication,'' *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9659–9672, Oct. 2021.

[24] S. Hussain, ''Efficient ray-tracing algorithms for radio wave propagation in urban environments,'' Doctoral dissertation, School Electron. Eng., Dublin City Univ., Dublin, Ireland, 2017.

[25] S. Hussain and C. Brennan, ''An efficient ray tracing method for propagation prediction along a mobile route in urban environments,'' *Radio Sci.*, vol. 52, no. 7, pp. 862–873, 2017.

[26] S. Hussain and C. Brennan, ''Efficient preprocessed ray tracing for 5G mobile transmitter scenarios in urban microcellular environments,'' *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3323–3333, May 2019.

[27] S. Hussain and C. Brennan, ''A visibility matching technique for efficient millimeter-wave vehicular channel modeling,'' *IEEE Trans. Antennas Propag.*, early access, Jun. 2, 2022, doi: 10.1109/TAP.2022.3178130.

[28] *COST 231 Munich Digital Map and Measurements*. Accessed: Jan. 9, 2022. [Online]. Available: https://propagationtools.com/wireless/cost-231-munich-digital-map-andmeasurements/

[29] L. Noriega, ''Multilayer perceptron tutorial,'' School of Comput., Staffordshire Univ., Staffordshire, U.K., vol. 34, 2005.

[30] K. O'Shea and R. Nash, ''An introduction to convolutional neural networks,'' 2015, *arXiv:1511.08458*.

[31] Y. Guo, Y. Zhou, X. Hu, and W. Cheng, ''Research on recommendation of insurance products based on random forest,'' in *Proc. Int. Conf. Mach. Learn., Big Data Bus. Intell. (MLBDBI)*, Nov. 2019, pp. 308–311.

[32] *Welcome to Colab!*. Accessed: Apr. 24, 2022. [Online]. Available: https://colab.research.google.com/

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and X. Zheng, ''TensorFlow: System for large-scale machine learning,'' *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and E. Duchesnay, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

KHALIL AHMAD recieved the B.Sc. degree in electrical engineering from the University of Engineering and Technology Peshawar, Peshawar, Pakistan, in 2019, and the M.Sc. degree in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2022. He is currently working as a Research Assistant with the Smart Agri-Tech Laboratory, NUST Inter-disciplinary Cluster for Higher Education (NICHE). His research interests include application of machine learning techniques in radio communication, healthcare, and agriculture.

SAJJAD HUSSAIN (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology at Taxila, Taxila, Pakistan, in 2006, the M.Sc. degree in telecommunications engineering from the University of Liverpool, Liverpool, U.K., in 2008, and the Ph.D. degree in electronic engineering from Dublin City University, Dublin, Ireland, in 2017. From 2009 to 2013, he was a Technical and Test Engineer at Vodafone Automotive Ltd., Manchester, U.K. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan. His research interests include radio channel modeling for 5G and beyond networks.