

Interpretable AI-Based Large-Scale 3D Pathloss Prediction Model for Enabling Emerging Self-Driving Networks

Usama Masood¹, Graduate Student Member, IEEE, Hasan Farooq², Member, IEEE, Ali Imran³, Senior Member, IEEE, and Adnan Abu-Dayya⁴, Senior Member, IEEE

Abstract—In modern wireless communication systems, radio propagation modeling to estimate pathloss has always been a fundamental task in system design and optimization. The state-of-the-art empirical propagation models are based on measurements in specific environments and limited in their ability to capture idiosyncrasies of various propagation environments. To cope with this problem, ray-tracing based solutions are used in commercial planning tools, but they tend to be extremely time-consuming and expensive. We propose a Machine Learning (ML)-based model that leverages novel key predictors for estimating pathloss. By quantitatively evaluating the ability of various ML algorithms in terms of predictive, generalization and computational performance, our results show that Light Gradient Boosting Machine (LightGBM) algorithm overall outperforms others, even with sparse training data, by providing a 65% increase in prediction accuracy as compared to empirical models and 13x decrease in prediction time as compared to ray-tracing. To address the interpretability challenge that thwarts the adoption of most Machine Learning (ML)-based models, we perform extensive secondary analysis using SHapley Additive exPlanations (SHAP) method, yielding many practically useful insights that can be leveraged for intelligently tuning the network configuration, selective enrichment of training data in real networks and for building lighter ML-based propagation model to enable low-latency use-cases.

Index Terms—Radio propagation model, mobile networks, pathloss, ray tracing, feature engineering, interpretable machine learning, explainable artificial intelligence (XAI), LightGBM, ultra-reliable low-latency communication (URLLC)

1 INTRODUCTION

EMERGING cellular networks are anticipated to witness a dramatic growth in connected devices and exciting new vertical services. Artificial Intelligence (AI) enabled end to end network automation vis-a-vis next generation Self Organizing Network (AISON), as proposed in [2], is considered to be the key enabler to meet the stringent performance requirements of increasingly complex planning, operation, optimization and maintenance in emerging self-driving networks.

The ultimate goal of AISON is to autonomously orchestrate the plethora of network parameters to maintain optimal multi-faceted network performance in terms of all important top level Key Performance Indicators (KPIs), without much human involvement [2]. Most of the high level KPIs such as capacity, Quality of Service (QoS) and

energy efficiency ultimately depend on one core low level metric i.e., Received Signal Strength (RSS). Therefore, characterising RSS as a function of network parameters is the very first step towards optimally designing and operating a cellular network. Hence, a realistic propagation model that is sensitive to the variations in network parameters (e.g., tilt) and environment geography and can follow the spatio-temporal variation in the network will be the cornerstone of AISON enabled future cellular networks (5G and beyond).

The existing propagation models can be categorized into three classes: deterministic, empirical and semi-empirical [3], [4]; *deterministic models* are based on the principles of wave propagation that can be theoretically computed using Maxwell's equations. However, in practice approximate methods such as *ray tracing* are used to model signal propagation, by taking into account the interactions of rays with the environment and using the dominant ray path to calculate the pathloss. These models can be very accurate depending on the resolution of available topographical database, but unfortunately are *computationally inefficient*. On the other hand, *empirical* and *semi-empirical models* such as COST-Hata [5], Stanford University Interim (SUI) [6], Standard Propagation Model (SPM) [7] and ITU-R P.452-15 [8] can be efficiently computed. However, these empirical and semi-empirical models are less sensitive to the actual physical and geometric structure of a given propagation environment and require in-depth domain-specific knowledge and technical expertise in radio signal propagation across electromagnetic fields.

- Usama Masood and Ali Imran are with AI4Networks Research Center, School of Electrical & Computer Engineering, University of Oklahoma, Tulsa, OK 74135 USA. E-mail: {usama.masood, ali.imran}@ou.edu.
- Hasan Farooq is with Ericsson Research, Santa Clara, CA 95051 USA. E-mail: hasan.farooq@ericsson.com.
- Adnan Abu-Dayya is with Electrical Engineering Department, Qatar University, Doha, Qatar. E-mail: adnan@qu.edu.qa.

Manuscript received 21 Oct. 2020; revised 9 July 2021; accepted 24 Jan. 2022. Date of publication 31 Jan. 2022; date of current version 5 June 2023.

This work was supported in part by National Science Foundation (NSF) under Grants 1559483, 1619346, and 1730650, and in part by Qatar National Research Fund (QNRF) under Grant NPRP12-S 0311-190302.

(Corresponding author: Usama Masood.)

Digital Object Identifier no. 10.1109/TMC.2022.3147191

To address the constraints and limitations of traditional propagation modeling methods, Artificial Intelligence and ML techniques are being considered as promising viable solutions and have been proven to be very effective for approximating arbitrary functions with hidden features. As envisioned in [2], AI is going to be indispensable in optimally designing and operating increasingly complex cellular networks. Hence, AI can replace classical mathematical models with a robust data-driven pathloss prediction model, that is more accurate than empirical propagation models and more computationally efficient than deterministic models, for system level intelligent network planning and post-deployment optimization and automation in cellular networks.

1.1 Related Work

In recent years several studies have been conducted for pathloss prediction in a particular environment using machine learning based models. Artificial Neural Networks (ANNs) have been at the core of most ML-based pathloss prediction models, however, the input features to the ANN model in these studies are limited to a particular environment, such as rural [9], urban [10], [11] and volcanic ocean islands [12], and seems unable to scale to other environment settings. The authors in [13] went one step ahead and used evolutionary algorithms to find the optimal hyper-parameters of the ANN based model, but they assumed a uniformly structured simulation area, which is not the case in practical scenarios. The authors in [14] incorporated features based on clutter maps to differentiate between different environments, but the presented model is still unable to capture the variation in coverage due to the change in geometrical structure of the propagation path. Furthermore, the authors in [15] tried to capture this variation by incorporating clutter heights as input feature, but still their input feature set is very limited to scale to different network configuration, as is the case in real networks. On the other hand, supervised ML techniques have also been used for pathloss prediction. The authors in [16] used Support Vector Regression to predict pathloss. However, they trained their model on a drive test data from a single serving Base Station (BS) in an urban environment. The authors in [17] compared the performance of several supervised ML algorithms for estimating cellular network coverage, using User Equipment (UE) measurement traces, BS parameters and geographical information. However, instead of modeling the pathloss or RSS, the authors classify the observation area as a good or a bad coverage area, using a pre-defined coverage threshold. A recently proposed data-driven model in [18] is the most relevant to our framework, using a boosting ensemble learning method to predict RSS using UE data from crowdsourcing applications. However, the environmental features used in the model are very limited.

1.2 Contributions and Organization

In this paper, we present a framework for an AI-driven large-scale 3D pathloss model (See Fig. 1), that is scalable and robust to the variations in the environment geography and addresses the limitations of aforementioned studies.

The contributions of this paper can be summarized as follows:

- 1) A novel set of key predictors (features) are identified that can characterize the physical and geometric structure of the environment traversed by a signal in its propagation path (e.g., *indoor distance*, *Manhattan distance*, *number of building penetrations in each clutter type*).
- 2) Multi-faceted performance comparison of the current state-of-the-art ML algorithms is done to evaluate the ability of proposed ML-based propagation model for real-time implementation, that includes predictive performance, generalization performance (robustness to unseen propagation scenarios using sparse training data, as is the case in real networks) and computational performance (i.e., training time and prediction time). In our investigation, Light Gradient Boosting Machine (LightGBM) algorithm is found to be the most optimal choice overall in modeling the complex propagation environment in real networks, due to its lightning fast training process and robustness to sparsity of training data.
- 3) The baseline performance of LightGBM algorithm is further optimized by investigating four different hyperparameter optimization approaches. These include Grid Search, Random Search, Bayesian Optimization and Simulated Annealing. These approaches are investigated in terms of performance gain and computational complexity (convergence time), and Bayesian optimization seems to be the best approach among them, as it converges in just 3 search iterations and reduces the prediction RMSE by 10%.
- 4) Performance comparison of the proposed model with state-of-the-art empirical propagation models and ray-tracing approach is also provided. The results show a 65% increase in prediction accuracy as compared to empirical propagation models and 13x decrease in prediction time as compared to ray tracing.
- 5) A key caveat of using ML is the lack of interpretability of resultant models, i.e., the black box paradox. In this study we try to address this weakness of the proposed RSS estimation models by performing extensive secondary analysis of the proposed models through SHAP method to interpret the model's predictions and bring clarity in understanding the importance of each feature (e.g., Azimuth, Tilt and Antenna Height) in the model. Utility of the insights drawn from the secondary SHAP analysis are also provided, such as intelligent optimization of network configuration, smart/selective enrichment of training data in real networks and building lighter ML-based propagation model to enable low-latency use-cases.
- 6) Another key contribution of the paper is to leverage the SHAP interpretability analysis to improve the various aspects of performance in the baseline model. We leverage the insights from SHAP analysis to propose a second novel light weight model for real-time implementation. This second model uses only the most significant features (selected based on insights gained from SHAP analysis) and as a result

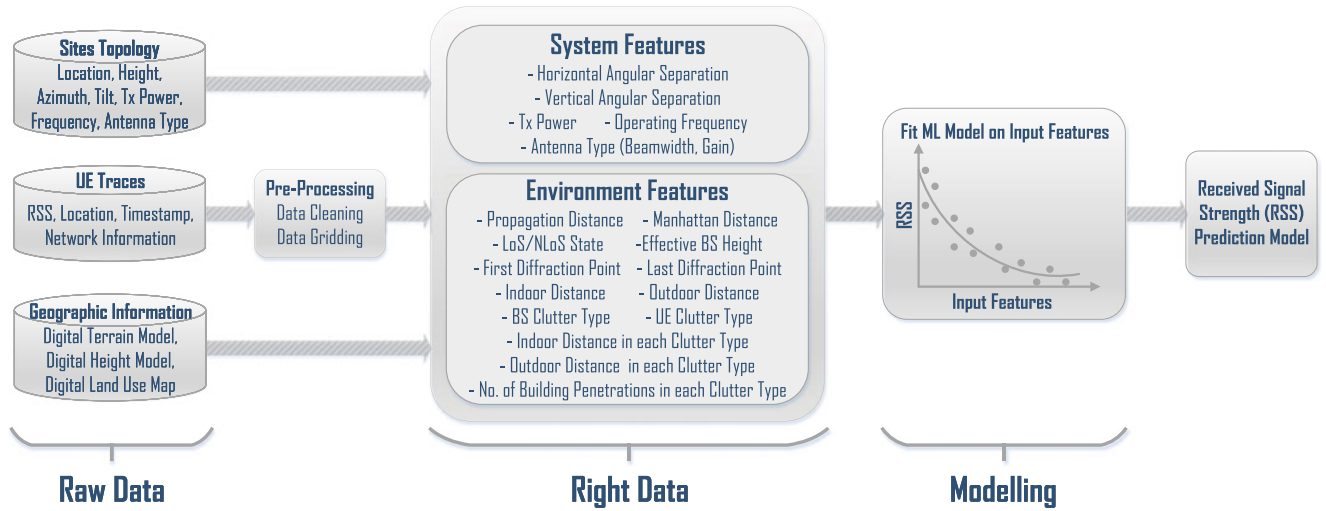


Fig. 1. Proposed framework of a machine learning-based 3D radio propagation model. Raw datasets such as BS sites topology, UE measurement traces and geographical information are first pre-processed and then converted into right data i.e- feature matrix comprising of key features/predictors that can estimate pathloss (or RSS). These key features are then used to train state-of-the-art ML algorithms for creating an RSS prediction model.

has ~ 70% less computational complexity compared to the base line model at the cost of negligible loss in performance (around 3%).

The rest of the paper is organized as follows: Section 2 explains the proposed framework, starting from raw data from the network, data pre-processing, feature engineering and a comparison of various machine learning algorithms for modeling RSS, whereas, Section 3 provides the performance comparison of the proposed model with traditional propagation models. Interpretation of the proposed AI-driven model using a recently proposed sensitivity analysis technique is given in Section 4 and finally Section 5 concludes this paper.

2 PROPOSED FRAMEWORK

The proposed framework for an AI-driven 3D propagation model (Fig. 1) uses raw data from the network consisting of network topology information, UE measurement traces and geographic information of the area, pre-process them and converts them into right data [2], which is then fed to a ML model to estimate RSS at given locations in a radio propagation environment.

2.1 Network and Simulation Setup

- **Network Scenario:** A ray-tracing based industry standard radio network planning and optimization platform “Atoll” [19] is used to create a sophisticated network topology, consisting of 10 macro cell sites in the center of City of Brussels, Belgium (See Fig. 2b). Actual antenna pattern and antenna heights are used in our analysis. Table 1 lists all the network and simulation scenario settings used in our analysis.
- **Geographical Datasets:** High resolution (1-m) geographical datasets containing earth terrain, buildings heights and land use information (e.g., Open, Parks, Block Buildings etc.) of the city are used in our analysis, enabling realistic and accurate pathloss calculation using Aster high-performance propagation model [20].
- **Ray Tracing:** Aster utilizes advanced ray-tracing propagation techniques to calculate various phenomena that affects radio wave propagation including vertical diffractions over roof-tops, horizontal diffractions and reflections based on ray-launching, atmospheric absorptions, rain attenuation and vegetation through loss etc. Aster utilizes advanced ray-

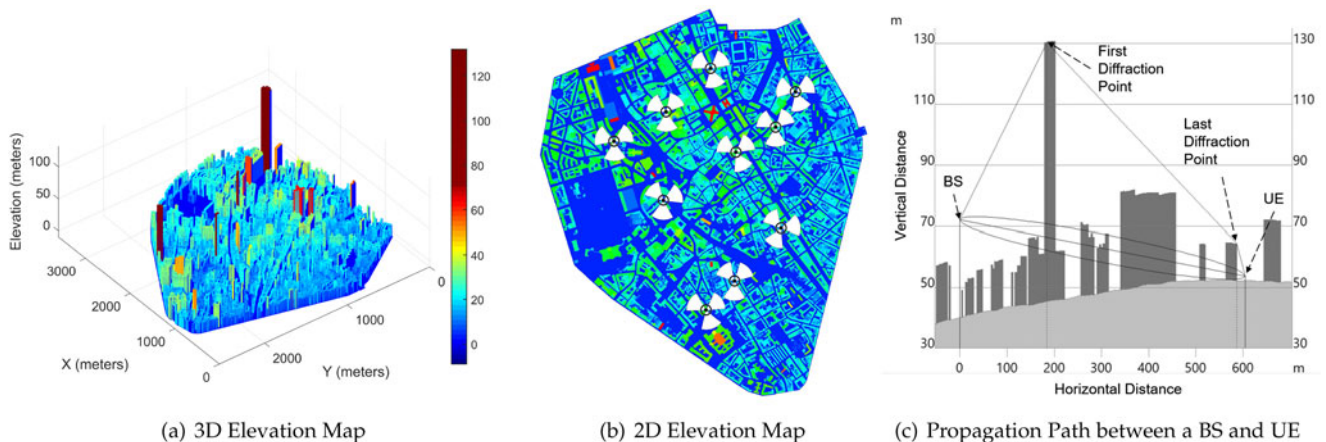


Fig. 2. Area of simulation showing (a) Building heights (b) Transmitter positions and (c) Vertical propagation path.

TABLE 1
Network Scenario Settings

System Parameters	Values
Air Technology	4G LTE
Cellular Layout	10 Macrocell sites
Sectors	3 sectors per eNB
Simulation Location	Brussels, Belgium
Simulation Area	3.80 km ²
User Distribution	Poisson Distribution
Propagation Model	Aster (Ray Tracing)
Path Loss Matrix Resolution	10m
Geographic Information	(1-m Resolution GeoData) Ground Heights (DTM) + Building Heights (DHM) + Land Use Map (DLU)
Land Cover (Clutter) Types	15 different classes
eNB Transmit Power (max)	43 dBm
eNB Noise Figure	5 dB
eNB Antenna Height	Actual site heights
eNB Antenna Model	Kathrein Directional Antenna (Model 742 265)
eNB Antenna Gain	18.3 dBi
eNB Antenna Horizontal Half Power Beamwidth	63°
eNB Antenna Vertical Half Power Beamwidth	4.7°
Frequency Band	2110 FDD (E-UTRA Band 1)
Channel Bandwidth (CBW)	5 MHz

tracing propagation techniques to calculate various phenomena that affects radio wave propagation including vertical diffractions over roof-tops, horizontal diffractions and reflections based on ray-launching, atmospheric absorptions, rain attenuation and vegetation through loss etc.

- **Validation using Real Data:** Furthermore, the parameters of the Aster propagation model are pre-calibrated using more than 1.5 million real channel measurement points from the real environment [21]. Therefore, the high-fidelity RSS (or pathloss) data calculated by Atoll in the observation area can be taken as ground truth for designing a realistic propagation model.

2.2 Raw Data

The following three different kinds of datasets are required as input data for our proposed framework:

- 1) **Sites Topology:** This dataset contains the Location, Height, Azimuth, Tilt, Transmit Power, Frequency,

Antenna Type of all the BSs in the observation area. It is denoted by D_{BS} (See Table 2).

- 2) **Geographic Information:** The geographical information of the propagation environment can be captured using three types of geographical datasets: Digital Terrain Model (DTM), Digital Height Model (DHM) and Digital Land Use Map (DLU). These datasets are in a raster grid format, which means that the whole observation area is divided into grids (or bins), each grid containing a specific value (See Table 3). These geographical datasets are routinely used by mobile telecom industry for their planning and maintenance tasks, and can be acquired on demand [22].
- 3) **UE Traces:** This dataset contains the RSS, Location, Timestamp, Network ID of all the UEs in the observation area. It is denoted by D_{UE} (See Table 4). The mobile operators can readily use the data from Drive Tests, Minimization of Drive Tests (MDT) reports, crowdsourcing applications etc. to generate this dataset, without the need for any new standardization.

2.3 Data Pre-Processing

Raw UE traces from the network are first pre-processed by cleaning and gridding, before using them for feature extraction.

2.3.1 Data Cleaning

Data cleaning is the process of identifying missing and corrupt values in the dataset and then handling them by modifying or deleting the relevant rows (or entries) from the dataset. This pre-processing step ensures that the training data for the proposed model is free from any anomalies and inconsistencies. In our study, some UE traces containing missing values (e.g., out of coverage UEs) are removed before using them for further analysis.

2.3.2 Data Gridding

Data gridding is the process of mapping all UE traces into unique spatial bins (of 10m width in our case) and then averaging the measurements inside each spatial bin for every serving BS. The advantage of data gridding is twofold:

- 1) **Handling Randomness:** First, it can offset randomness in RSS due to fast fading and slow fading (to some extent), by averaging all measurements from the same BS, falling within a small bin, given the RSS

TABLE 2
Sites Topology

Parameter	Description
Location (x_{BS}, y_{BS})	Location coordinates of a BS site (See Fig. 2b)
Height (h_{BS})	The height of BS antenna above the ground and building (if any)
Azimuth (θ_{BS})	Azimuth angle (in degrees) of the BS antenna, which is the direction of antenna w.r.t. North
Tilt (ϕ_{BS})	Tilt angle (in degrees) of the BS antenna, which is basically the angle below the horizontal plane
Tx Power (P_{BS})	The power of the radio signal (in dBm) when it's transmitted from the BS antenna
Frequency (f)	Carrier frequency used by the BS antenna for transmission
Antenna Type	The type of antenna used by the BS transmitter. It is differentiated by beamwidth, antenna gain etc

TABLE 3
Geographic Information

Parameter	Description
Digital Terrain Model (D_{DTM})	It provides the earth terrain (ground) height. It takes as an input the x, y coordinates and outputs the ground height z_{DTM} at that place $z_{DTM} = D_{DTM}(x, y)$. (See Fig. 2c)
Digital Height Model (D_{DHM})	It provides the building heights (above the ground) in the observation area. It takes as an input the x, y coordinates and output the total height z_{DHM} of the user at that place $z_{DHM} = D_{DHM}(x, y)$. (See Fig. 2a)
Digital Land Use Map (D_{DLU})	It provides the clutter (or land cover) type of each grid in the observation area. It takes as an input the x, y coordinates and output the clutter type c at that place $c = D_{DLU}(x, y)$

within the bin is expected to stay almost same due to its small size.

- 2) **Handling Positioning Error:** Second, it handles positioning error in the user reported measurements. However, gridding/binning has its costs i.e., it introduces quantization error to say the least and also presents an accuracy versus complexity trade-off.

In our study, UE traces in the observation area are first mapped into unique spatial bins of 10m width, and then in each bin all UE traces corresponding to a unique BS were averaged out. For further analysis, the averaged UE traces are used to mitigate the effect of randomness from the original data.

For detailed analysis of the impact of gridding, reader is referred to a recent study in [23] and [24]. Analysis presented in [23] and [24] shows that there exists a trade-off between the quantization error introduced by gridding and the positioning error from the incorrect GPS location tagged with the UE measurements, and that there exists an optimal bin-width for gridding for a given UE density and positioning error that maximizes the accuracy of UE measurements data.

2.4 Feature Engineering (Raw Data to Right Data)

Feature engineering is a key process in ML, that leverages domain knowledge to create features which can characterize the complex target model and greatly enhance its learning performance.

In our study, several *key predictors (features)* are identified or engineered, to better characterize the environment traversed by a signal in its propagation path (See Fig. 3). The raw network, UE and geographic datasets, readily available to the mobile operators, are converted into right data (key features) comprising of system as well as propagation environment features that can then be leveraged to train an ML-based propagation model. The list of symbols used in this paper is given in Table 5.

TABLE 4
UE Traces

Parameter	Description
RSS	Received Signal Strength (P_{UE}) from the serving Base Station (BS)
Location	Location coordinates (x_{UE}, y_{UE}) of a UE
Timestamp	Time at which the UE trace is recorded
Network ID	Information regarding serving BS ID, Mobile Network Code etc.

2.4.1 Propagation Distance

This is the horizontal distance (in meters) between a UE and its serving BS. It is denoted by d

$$d = \sqrt{(x_{BS} - x_{UE})^2 + (y_{BS} - y_{UE})^2}. \quad (1)$$

2.4.2 Horizontal Angular Separation

This is the horizontal angular separation (in degrees) between the BS antenna boresight and the direction of Line of Sight path to the UE. This feature captures the attenuation due to horizontal antenna pattern of the BS. It is denoted by θ_{hor}

$$\theta_{hor} = \text{abs}(\theta_{BS} - \theta_{UE}), \quad (2)$$

where

$$\theta_{UE} = \text{atan2}\left(\frac{x_{BS} - x_{UE}}{y_{BS} - y_{UE}}\right). \quad (3)$$

Here θ_{UE} is the azimuth angle of (UE) arrival and θ_{BS} is the azimuth angle of (BS) departure, or simply BS azimuth angle, whereas, $\text{atan2}()$ calculates the four quadrant inverse tangent.

2.4.3 Vertical Angular Separation

This is the vertical angular separation (in degrees) between the BS antenna boresight and the direction of Line of Sight path to the UE. This feature captures the attenuation due to vertical antenna pattern of the BS. It is denoted by ϕ_{ver}

$$\phi_{ver} = \phi_{UE} - \phi_{BS}, \quad (4)$$

where

$$\phi_{UE} = \text{atan}\left(\frac{z_{UE} - z_{BS}}{d}\right), \quad (5)$$

$$z_{UE} = D_{DTM}(x_{UE}, y_{UE}) + D_{DHM}(x_{UE}, y_{UE}), \quad (6)$$

$$z_{BS} = D_{DTM}(x_{BS}, y_{BS}) + D_{DHM}(x_{BS}, y_{BS}). \quad (7)$$

Here ϕ_{UE} is the tilt angle of (UE) arrival, ϕ_{BS} is the tilt angle of (BS) departure, or simply BS tilt angle, z_{UE} is the total height of UE and z_{BS} is the total height of BS (See Table 3 for details on D_{DTM} and D_{DHM}).

2.4.4 Effective BS Height

This is the vertical distance (in meters) between a UE and its serving BS. It is denoted by d_{vert}

$$d_{vert} = z_{BS} - z_{UE}. \quad (8)$$

2.4.5 Manhattan Distance

This represents the Manhattan distance between the BS and UE. As radio waves also diffracts at the street corners and

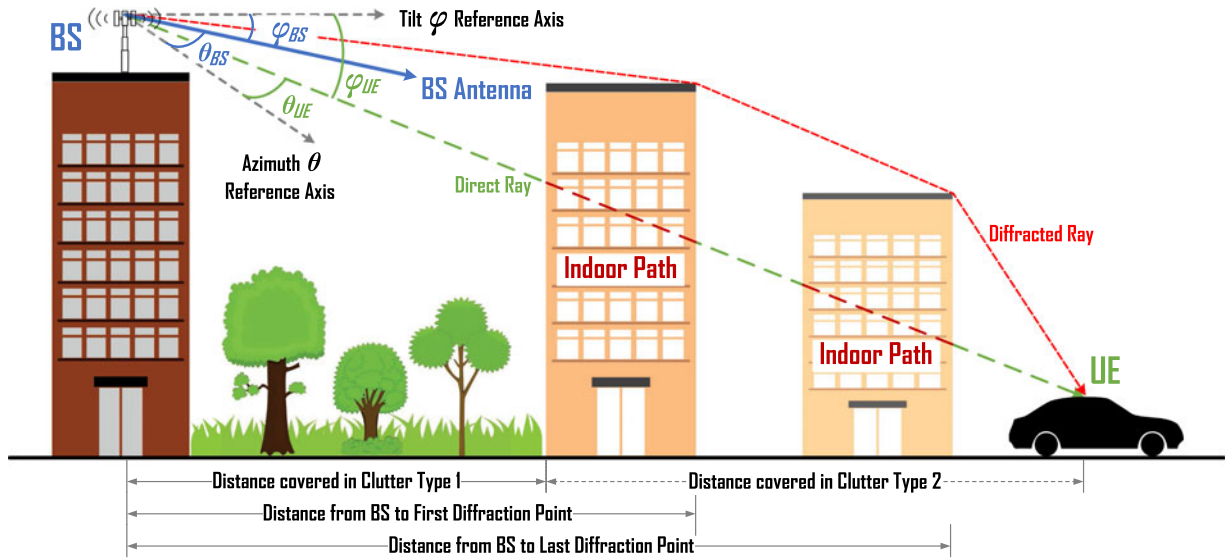


Fig. 3. Propagation path between a BS and UE, showing various features of the proposed model.

are more likely to travel along the streets in urban areas, therefore Manhattan distance is a better metric to calculate the distance traversed by the signal, especially in urban networks [25], [26]. It is denoted by d_{man} .

2.4.6 LoS/NLoS State

This represents the link status between a BS and a UE antenna. A UE can either be in a Line of Sight (LoS) or Non Line of Sight (NLoS) region from the BS. This feature is particularly useful in wireless channels, as LoS regions have higher RSS, and vice versa. It is denoted by L .

2.4.7 First Diffraction Point

This is the horizontal distance (in meters) from a BS to the first diffraction point in the propagation path between a BS and UE. This feature captures the significance of diffracted rays at the receiver, as multiple rays from the same BS are received and the ray having the highest signal strength is selected as the dominant ray. It is denoted by d_{FD} .

2.4.8 Last Diffraction Point

This is the horizontal distance (in meters) from a BS to the last diffraction point in the propagation path between a BS and UE. This feature also tries to learn the behavior of diffracted rays in the estimation of RSS. It is denoted by d_{LD} .

2.4.9 Number of Building Penetrations

This is the number of buildings penetrated by the signal in its direct path between a BS and UE. This feature characterizes the penetration loss (dB) experienced by the signal while crossing buildings. It is denoted by N .

2.4.10 Indoor Distance

Horizontal distance (in meters) in the direct path between a BS and UE that is passing through buildings (indoor). This feature characterizes the linear loss (dBm/m) experienced by the signal in an indoor area. It is denoted by d_{indoor} .

TABLE 5
Key Symbol Definitions

Symbol	Units	Definition	Symbol	Units	Definition
θ_{BS}	$^\circ$	Azimuth Angle of (BS) Departure	P_{BS}	dBm	BS Transmit Power
ϕ_{BS}	$^\circ$	Tilt Angle of (BS) Departure	P_{UE}	dBm	RSS of a UE
θ_{UE}	$^\circ$	Azimuth Angle of (UE) Arrival	d	m	Propagation Distance
ϕ_{UE}	$^\circ$	Tilt Angle of (UE) Arrival	d_{vert}	m	Effective BS Height
ϕ_{ver}	$^\circ$	Vertical Angular Separation	d_{man}	m	Manhattan Distance
θ_{hor}	$^\circ$	Horizontal Angular Separation	L	-	LoS/NLoS State
f	MHz	Operating Frequency	d_{FD}	m	Distance from BS to First Diffraction Point
h_{BS}	m	BS Antenna Height	d_{LD}	m	Distance from BS to Last Diffraction Point
h_{UE}	m	UE Antenna Height	N	-	Number of Building Penetrations
c_{BS}	int	BS Clutter Type	d_{indoor}	m	Indoor Distance in the Propagation Path
c_{UE}	int	UE Clutter Type	$d_{outdoor}$	m	Outdoor Distance in the Propagation Path
D_{DHM}	-	Raster Grid Data of Digital Height Model	N_c	-	No. of Building Penetrations in each Unique Clutter
D_{DTM}	-	Raster Grid Data of Digital Terrain Model	d_{indoor_c}	m	Indoor Distance in each Unique Clutter
D_{DLU}	-	Raster Grid Data of Digital Land Use Map	d_{indoor_e}	m	Outdoor Distance in each Unique Clutter

2.4.11 Outdoor Distance

Horizontal distance (in meters) in the direct path between a BS and UE that is in open area (outdoor). This feature characterizes the linear loss (dBm/m) experienced by the signal in an open area. It is denoted by $d_{outdoor}$.

2.4.12 BS Clutter Type

It is the clutter type (or land cover type) of the BS (For example: open street, dense buildings, sparse buildings, trees, water etc.). This feature tries to learn the effect of different land cover type on the signal around the BS antenna. It is denoted by c_{BS} .

2.4.13 UE Clutter Type

It is the clutter type (or land cover type) of the UE (For example: open street, dense buildings, sparse buildings, trees, water etc.). Each clutter type has its own effect on the signal and this feature tries to learn this behavior. It is denoted by c_{UE} .

2.4.14 Number of Building Penetrations in Each Clutter Type

This is the number of buildings penetrated by the signal in each unique clutter in the direct path between a BS and UE. Different clutters can be different types of buildings, each having different penetration loss (dB). If our observation area consists of 15 different clutter classes, then this feature is subdivided into 15 different features, each representing the number of building penetrations in that respective clutter, whose sum equals the total number of building penetrations in the propagation path of that UE. It is denoted by N_c .

2.4.15 Indoor Distance in Each Clutter Type

Indoor distance (in meters) covered by each unique clutter in the direct path between a BS and UE. This feature characterizes the linear loss (dBm/m) experienced by the signal in different indoor environments. Again, this feature is subdivided into the total number of clutters in the observation area, whose sum equals the total indoor distance in the propagation path of that UE. It is denoted by d_{indoor_c} .

2.4.16 Outdoor Distance in Each Clutter Type

Outdoor distance (in meters) covered by each unique clutter in the direct path between a BS and UE. This feature characterizes the linear loss (dBm/m) experienced by the signal in different outdoor environments. Again, this feature is subdivided into the total number of clutters in the observation area, whose sum equals the total outdoor distance in the propagation path of that UE. It is denoted by $d_{outdoor_c}$.

2.5 RSS Modeling Using Machine Learning Methods

RSS prediction is essentially a regression problem, where the key features proposed earlier are used as input for training ML models, to learn the complex behavior of a signal passing through a wireless channel. Algorithm 1 explains the process of removing randomness from the UE traces (to some extent) by gridding (averaging all measurements in a spatial bin) and then training the ML model using the

computed key features as input and the corresponding expected value of RSS as output.

In this paper, we investigate a range of different *Parametric*, *Non-Parametric* and Ensembles of machine learning regression algorithms for their strengths and weaknesses while modeling RSS and implementing them.

- Parametric algorithms, such as Linear Regression, assumes training data to be of a specific functional form with a fixed size of parameters.
- Non-Parametric algorithms, on the other hand, such as k-Nearest Neighbors, Decision Tree and Neural Networks, are free to assume any functional form of the training data.
- Ensemble algorithms are of two types: Bagging and Boosting, which further have several variants.

Earlier works on propagation modeling [17] were mostly based on parametric models and some ensemble learning models.

Algorithm 1. RSS Prediction Model Training Algorithm

Input: $D_{UE}, D_{BS}, D_{DTM}, D_{DHM}, D_{DLU}$

Output: RSS Prediction Model $\mathcal{M}(F)$

- 1: **for** all UE traces **do**
 - 2: Map its location to pre-defined grids (e.g., 10m x 10m)
 - 3: **end for**
 - 4: **for** each unique grid **do**
 - 5: **for** each unique serving BS **do**
 - 6: Average out the RSS (P_{UE}) of all users to offset randomness
 - 7: Compute feature vector $F = [d, \theta_{hor}, \phi_{ver}, d_{vert}, d_{man}, L, d_{FD}, d_{LD}, N, d_{indoor}, d_{outdoor}, c_{BS}, c_{UE}, N_c, d_{indoor_c}, d_{outdoor_c}]$
 - 8: **end for**
 - 9: **end for**
 - 10: Train the Machine Learning model \mathcal{M} using Feature Matrix \mathcal{F} , whose each row corresponds to a feature vector F
 - 11: **return** $\mathcal{M}(F)$
-

2.5.1 Criteria for Model Evaluation and Selection

In this work, we have done a comprehensive and multi-faceted performance evaluation of the state-of-the-art ML algorithms, that includes *predictive performance*, *generalization performance* and *computational performance*, and also provided insights from each of these algorithms to make this paper self-contained.

- 1) Predictive Performance: The predictive performance of a model indicates the model accuracy for unseen data. In our analysis, we have used Root Mean Square Error (RMSE) and coefficient of determination (R^2) performance metrics defined below to judge the predictive performance of models

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (P_{UE_i} - \hat{P}_{UE_i})^2}{N}}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} = 1 - \frac{\sum_i (P_{UE_i} - \hat{P}_{UE_i})^2}{\sum_i (P_{UE_i} - \bar{P}_{UE_i})^2}$$

Here P_{UE} is the actual RSS of a UE, \hat{P}_{UE} is the predicted RSS of a UE, \bar{P}_{UE} is the mean value of RSS and N is the number of UE traces in the test data. SS_{res} and SS_{tot} corresponds to the *residual sum of squares* and *total sum of squares*, respectively. $RMSE$ is measured here in dB, whereas $R^2 = 1$ in the best case and $R^2 = 0$ when the model output is always equal to its mean value in test data. In rare scenarios, $R^2 < 0$, when the model predictions are even worse than the baseline mean value prediction.

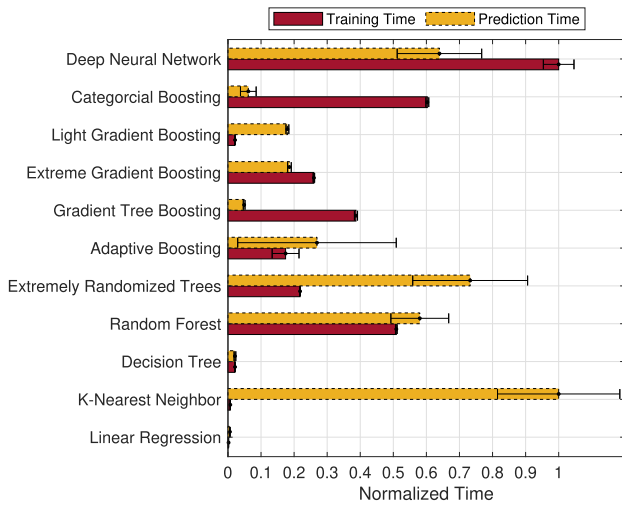
- 2) Generalization Performance: The generalization performance of a model indicates its robustness in predictive performance for unseen data (or scenarios). In our analysis, we have used 5-fold repeated cross-validation technique along with its mean and standard deviation value for all iterations to judge the generalization ability of a model on unseen data (propagation scenarios) with a certain confidence, even when using very sparse training data (2% in our analysis). In other words, the generalization ability of the model can be judged both by the standard deviation of its mean RMSE or the increase in its predictive performance when using sparse training data for all cross-validation iterations.
- 3) Computational Performance: Computational performance of a ML model can be judged by its training time, which indicates the time and therefore resources it takes to train the model and prediction time, which shows its prediction latency. These values are extremely crucial in a production setting where we have cost (or resources) and latency constraints. In our analysis, all the ML methods are evaluated using the same number of CPU cores for a fair comparison.

2.5.2 Model Evaluation

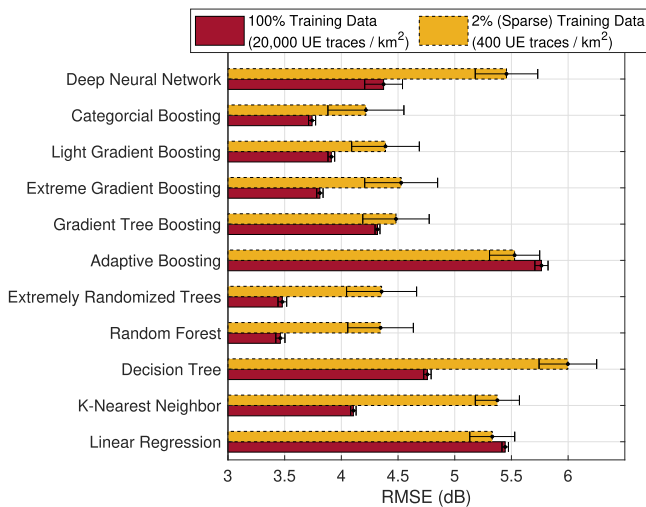
- 1) Linear Regression: As evidenced by our experiments, linear regression method [27] doesn't seem to be suitable to capture the complex non-linear nature of the wireless channel. In our results, we have shown (in Fig. 4) that it gives a high prediction Root Mean Square Error (RMSE) of 5.45 dB and a low coefficient of determination (R^2) score of 0.65.
- 2) k-Nearest Neighbors: k-Nearest Neighbors (k-NN) [28] doesn't seem to handle non-linearity well in case of sparse training data, as the prediction in test data is basically the mean of k nearest data points in the training data. Also, it has the highest computation cost at run-time among the tested algorithms, as evidenced in results (Fig. 4).
- 3) Decision Tree: A single Decision Tree (DT) [29] in our experiments, is unable to generalize well, especially with sparse training data and seems to overfit, therefore also doesn't seem to be a suitable choice for a ML-based propagation model.
- 4) Random Forest: Random Forest [30] is an Ensemble learning method, which combines several decision trees, using *Bootstrap Aggregation (or Bagging)* technique, to improve the predictive performance of a single decision tree. Here, each tree is trained on a random subset, with replacement, of training data.

Also, each node is split among a random subset of input features, which may not be the best split among all features. This randomness increases the bias of the forest, when compared to a single non-random tree. The output here is the average prediction of all individual trees, and due to this averaging, variance in the ensemble model decreases, which more than make up for the increase in bias, hence improving performance of the overall model, $RMSE$ of 3.46 dB as compared to 4.76 dB in a single decision tree. Another advantage is that, as opposed to a single decision tree, random forest is robust to outliers in the training data. The main drawback of using this method is the slow prediction speed, as evidenced in our results (See Fig. 4), due to a large number of trees, making it unsuitable for real-time predictions.

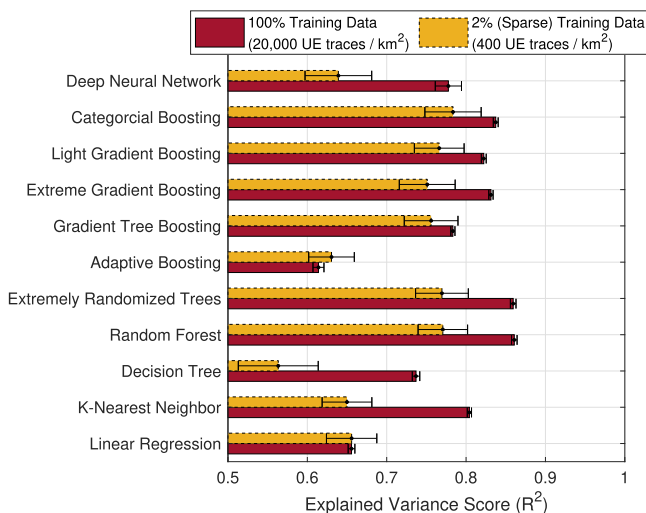
- 5) Extremely Randomized Trees: Extremely Randomized Trees is another *Bagging Ensemble* learning method, which goes one step further in randomizing the trees, as compared to Random Forest. In addition to each tree trained on a random subset of data and best split at each node chosen on a random subset of features, thresholds are also picked at random for every candidate feature at a node. This increase in randomness, further decreases the variance of the ensemble model, at the cost of slight increase in bias. It has all the pros of Random Forest, plus a reduction in training time, but the major drawback is still high prediction time, as evidenced in our results (See Fig. 4).
- 6) Adaptive Boosting: Adaptive Boosting (AdaBoost) is a *Boosting Ensemble* model. In boosting, models are built in sequence, so that each subsequent model learns from the mistakes of the previous model, to create a stronger model. In AdaBoost, each subsequent model is forced to focus on samples which were badly predicted in the previous model. This is done by giving higher weights to those samples in the training set, whose prediction error was high in the previous model, and vice versa. Weighted sampling is then used in the subsequent model to generate a derived training set, using sampling with replacement. The probability that a training example appears in the training set is relative to its weight. The final output is a weighted average of all the model's output, where more weight is placed on stronger models. Consequently, *the bias of the combined model is reduced in boosting, as opposed to bagging, where the variance was reduced*, by averaging several weak learners. As shown in Fig. 4, bagging methods outperforms AdaBoost in terms of higher prediction accuracy. The other disadvantage of this technique is that it cannot be parallelized, since the training of each subsequent tree model, depends on the output of previous model, therefore, its training time is much higher as compared to bagging ensemble methods.
- 7) Gradient Boosting Decision Trees: Gradient Boosting Decision Trees (GBDT) is another Boosting ensemble model, which works on the same Boosting principle



(a) Training time (normalized by the highest value of DNN) and Prediction Time (normalized by the highest value of k-NN)



(b) RSS Prediction Error



(c) Model R-Squared Value

Fig. 4. Comparison of various machine learning algorithms w.r.t (a) Training time, prediction time, (b) Prediction error, (c) R-Squared value and robustness to sparsity of training data, for modeling RSS (Height of bars represent the mean value and error bar represent the standard deviation using 5-fold repeated cross validation).

of learning from the previous model’s mistake, but the difference lies in how to learn from the mistakes of previous model. *Gradient Boosting learns from the error residuals (gradients) of the previous model directly*, unlike AdaBoost, which changes the sample distribution at every iteration, by giving higher weights to under-fitted (or badly predicted) samples. The goal is to iteratively minimize the prediction error, by training each subsequent decision tree model, on the residual errors (prediction errors) made by the previous model, this process is essentially a gradient descent optimization on the overall composite model. The final output would then be the sum of predictions from all the models. Our results show that Gradient Tree Boosting has much better prediction performance (*RMSE of 4.32 dB*) as compared to AdaBoost (See Fig. 4).

- 8) Extreme Gradient Boosting: Extreme Gradient Boosting (XGBoost) [31] is an advanced implementation of gradient boosting and follows the same principle. The main advantage of XGBoost is the ability of parallel processing, therefore much faster as compared to GBDT. While it’s not possible to create trees in parallel because each tree is dependent on the previous, it’s possible to build a tree using all the cores, by building several nodes within each depth of a tree, in parallel. To improve the performance of the model, *Weighted Quantile Stretch* idea is used to reduce the search space while finding the best split, by looking at the distribution of features across all instances in a leaf. It further reduces the computational complexity by learning the sparsity patterns in the data and skip samples with missing values while making a split. Moreover, it includes *regularization* to prevent over-fitting and improve overall performance of the model (See Fig. 4), due to which it is also sometimes called as *regularized gradient boosting*.
- 9) Light Gradient Boosting: Light Gradient Boosting Machine (LightGBM) [32] is another implementation of Gradient Boosting, which improves on XGBoost. It can train on larger datasets in a fraction of time and with comparable accuracy, as compared to XGBoost, hence the word *Light* is used. It uses a technique called *Gradient-based One-Side Sampling (GOSS)* to intelligently extract the most useful information as fast as possible, by randomly skipping the samples with less information (small gradients) in the dataset. Moreover, it has introduced another method called *Exclusive Feature Bundling (EFB)* for reducing model complexity, by combining similar features in a near lossless way. Our results have shown that LightGBM has better accuracy as compared to XGBoost in sparse training data scenario and 12x faster training speed (See Fig. 4).
- 10) Categorical Boosting: Categorical Boosting (CatBoost) [33] is a gradient boosting algorithm whose power lies in processing categorical features in the dataset. Categorical features have values which are discrete and not related to each other. CatBoost incorporates several innovative methods to deal with these features at training, instead of during

data pre-processing. Furthermore, it incorporates a modified gradient boosting algorithm called *ordered boosting*, to avoid *target leakage* present in standard gradient boosting algorithms, as they rely on the target of all training samples at each iteration, resulting in biasness. Here, however, training is done on independent random permutations of the dataset at each iteration, to avoid this *prediction shift*. Therefore, CatBoost can outperform other gradient boosting algorithms, specially if you have categorical variables in the data (for instance, LOS State, BS and UE Clutter Types are the categorical features used in our model). As shown in our results (Fig. 4), prediction RMSE is reduced to 3.74 dB at the cost of increase in training time. It is worth mentioning here that the GPU implementation of this algorithm is faster than both XGBoost and LightGBM, but in our results we have used CPU for training these algorithms.

- 11) **Deep Neural Network:** Deep Neural Network (DNN) algorithm belongs to a special class of machine learning, called *deep learning* and creates a *multi-layer perceptron (MLP)* to find the input-output associations. Its basic structure consists of an input layer, output layer and one or more hidden layers between them, each containing several neurons (or nodes). Neurons in the input layer equals the number of input features, whereas output layer consists of one neuron which holds the prediction output. Number of hidden layers and its neurons are variable, and depends on the complexity of model it is trying to learn. Our extensive investigations on DNN design show that, for learning the behavior of RSS in a wireless channel, 6 hidden layers each consisting of 32 neurons provide the most optimal results, any increase or decrease in this number results in over-fitting or under-fitting on the training data, respectively. The DNN used in this study has *Rectified Linear Unit (ReLU)* activation function in the hidden layers, whereas output layer uses *linear* activation function. In our simulation results (Fig. 4), DNN performs worse than ensemble-based methods and also has the highest computational cost.

2.5.3 Model Selection

For selecting the best performing model, we should overall look at the predictive, generalization and computational performance across all evaluated models. For a fair comparison, all the ML methods are evaluated using the same number of CPU cores. Furthermore, in all experiments, 5-fold repeated cross validation is used so that the results are generalizable in all propagation scenarios.

In Fig. 4a, training and prediction time of all the methods are normalized to the highest value individually. DNN has the highest, whereas linear regression has the lowest training and prediction time. In Fig. 4c, comparison of R^2 value is given, where CatBoost and LightGBM algorithms perform the best in capturing the variance of RSS (or pathloss) and learning complex non-linear relationships in a wireless channel and have the lowest Root Mean Squared Error (RMSE) in sparse training data scenarios (shown in Fig. 4b),

whereas linear regression has the highest RMSE for RSS prediction as the complex non-linear nature of wireless channel renders it unsuitable.

All the models are also separately trained on only 2% of training data, to evaluate their performance in case of data sparsity, as is the case in real practical scenarios. DNN shows the highest impact (loss in accuracy) due to data sparsity.

Overall, LightGBM algorithm outperforms others, especially for real-time implementation, due to its lightning fast training process. RSS model trained using LightGBM algorithm is used for further simulations and results.

2.5.4 Model Improvement Using Hyperparameter Optimization

The baseline LightGBM model performance shown in Fig. 4 is further improved by optimizing its hyperparameters for the RSS (or pathloss) prediction task, according to the special characteristics of the wireless channel.

The hyperparameters selected to be optimized are: 1) 'number of estimators' in the range of 500 – 2500, 2) 'maximum tree depth' in the range of 5 – 20 and 3) 'learning rate' in the range of 0.1 – 0.001. Furthermore, 5-fold repeated cross validation is used for each combination of hyperparameters and model's performance is evaluated based on its RMSE and R^2 . Four different hyperparameter optimization approaches are evaluated (shown in Fig. 5) in terms of performance gain and convergence time:

- 1) **Grid Search:** This approach does an exhaustive search over the entire search space of hyperparameters. Fig. 5a shows the mean RMSE and R^2 of LightGBM model at different combinations of hyperparameters. The model converges to its best RMSE after 50 iterations (as shown in Fig. 5e).
- 2) **Random Search:** This approach also does an exhaustive search over the entire search space of hyperparameters, but picks them randomly, therefore its convergence time is more likely to be less than grid search method, as shown in Fig. 5e, where the model converges to its best RMSE after 25 iterations.
- 3) **Bayesian:** In this approach, hyperparameters are tuned using a Bayesian optimization algorithm, known as Tree-structured Parzen Estimator (TPE) [34]. This Bayesian approach is a model-based approach, and as search iterations progresses, it switches from exploration to exploitation to minimize the objective function loss (concentrating on the hyperparameter combinations that resulted in lower loss, which in our case is the RMSE). This approach sometimes gets trapped in the local minima of the objective function, an issue which is not faced by grid or random search. Figs. 5c and 5e shows the superior performance of this approach as it converges in only 3 search iterations.
- 4) **Simulated Annealing:** This approach is a meta-heuristic optimization algorithm [35], that is simpler and is preferred over its Bayesian counterpart when the objective function is simple to evaluate.

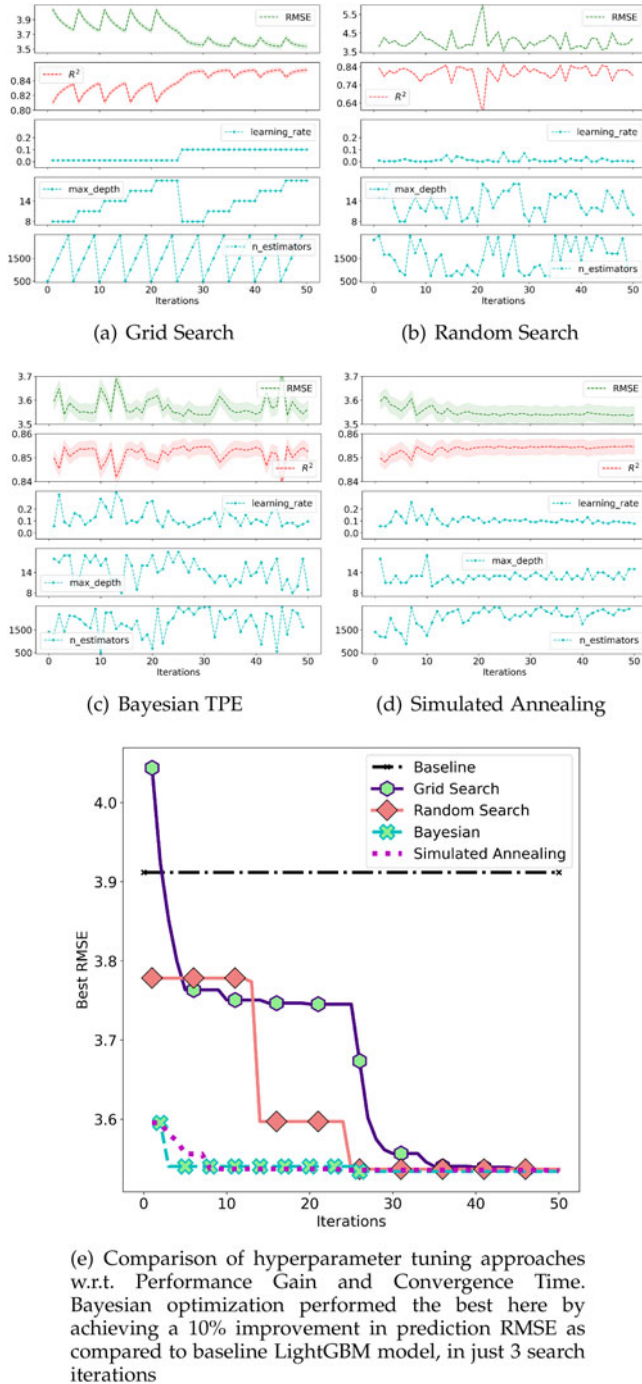


Fig. 5. Comparison of different Hyper-parameter tuning approaches for improving the performance of baseline LightGBM model, in terms of RSS Prediction RMSE, R^2 and convergence time (In subplots (a)-(d), RMSE and R^2 of the LightGBM model is plotted against different combinations of hyperparameters, after performing 5-fold repeated cross validation at each search iteration of the tuning process (dotted line/curve represents the mean value and filled area/polygon around it represents the standard deviation using 5-fold repeated cross validation).

But it seems to converge slowly than the Bayesian approach (as evidenced in Fig. 5e, where it took 8 iterations to converge). After these 8 iterations, our proposed ML algorithm shows a significant performance gain, as its prediction RMSE is reduced to 3.54 dB, as compared to 3.91 dB earlier in the baseline LightGBM algorithm using the default hyperparameters.

3 COMPARISON WITH EMPIRICAL RADIO PROPAGATION MODELS

We also compare the performance of our proposed AI-driven 3D propagation model based on improved LightGBM algorithm with traditional empirical propagation models, as they are currently used in state-of-the-art commercial planning tools to characterize the propagation behavior of a radio signal in different conditions. Empirical models offer a mathematical equation to calculate the path loss at any given point from the BS, and are based on data collected in a specific scenario.

3.1 COST-Hata Model

It is an empirical model for pathloss calculation [5], that extends the Hata formulae [36] to frequencies upto 2 GHz and it also takes into account the topo map (DTM) between the BS and UE and morpho map (DLU) only at the receiver. The below equation is valid for urban environments with 1.5 m UE height

$$L_{path} = A_1 + A_2 * \log(f) + A_3 * \log(h_{BS}) + B_1 + B_2 * \log(h_{BS}) + B_3 * h_{BS} * \log(d). \quad (9)$$

Here L_{path} is the pathloss (in dB), $A_1 = 46.3$, $A_2 = 33.9$, $A_3 = -13.82$, $B_1 = 44.9$, $B_2 = -6.55$, $B_3 = 0$ are user-defined parameters, f is the carrier frequency (in MHz), h_{BS} is the height of BS and d is the propagation distance between BS and UE.

For Urban Areas

$$L'_{path} = L_{path} - a(h_{UE}).$$

For Sub-Urban Areas

$$L'_{path} = L_{path} - a(h_{UE}) - 2 * \left(\log\left(\frac{f}{28}\right) \right)^2 - 5.4.$$

For Quasi-Open Rural Areas

$$L'_{path} = L_{path} - a(h_{UE}) - 4.78 * (\log(f))^2 + 18.33 * \log(f) - 35.94.$$

For Open Rural Areas

$$L'_{path} = L_{path} - a(h_{UE}) - 4.78 * (\log(f))^2 + 18.33 * \log(f) - 40.94.$$

Where L'_{path} is the corrected pathloss and $a(h_{UE})$ is the correction factor for UE height different from 1.5 m.

For Rural/Small Cities

$$a(h_{UE}) = (1.1 * \log(f) - 0.7) * h_{UE} - (1.56 * \log(f) - 0.8).$$

For Open Rural Areas

$$a(h_{UE}) = 3.2 * (\log(11.75 * h_{UE}))^2 - 4.97.$$

3.2 Stanford University Interim (SUI) Model

It is derived from the Erceg-Greenstein propagation model [37] and is valid for 1900-6000 MHz. It also takes into

account the topo map (DTM). It uses the following formula:

$$L_{path} = -7366 + 26 * \log(f) + 10 * a(h_{BS}) * (1 + \log(d)) - a(h_{UE}), \quad (10)$$

where

$$a(h_{BS}) = a - b * h_{BS} + \frac{c}{h_{BS}},$$

$$a(h_{UE}) = X * \log\left(\frac{h_{UE}}{2}\right).$$

Here $a(h_{BS})$ and $a(h_{UE})$ are the correction factors for BS and UE antenna heights, respectively, f is the operating frequency and d is the propagation distance (in km). $a = 4.6$, $b = 0.0075$, $c = 12.6$ and $X = 10.8$ are the correction constants which depend on the terrain type [6].

3.3 Standard Propagation Model (SPM)

It is derived from the Hata formulae and is valid for 150-3500 MHz. It also takes into account the topo map (DTM) and morpho map (DLU) between the BS and UE. It is given by the following formula:

$$L_{path} = K_1 + K_2 * \log(d) + K_3 * \log(h'_{BS}) + K_4 * L_{diff} + K_5 * \log(d) * \log(h'_{BS}) + K_6 * h'_{UE} + K_7 * \log(h'_{UE}) + K_{clutter} * f(clutter).$$

Here $K_1 = 23.8$, $K_2 = 44.9$, $K_3 = 10.89$, $K_4 = 0.19$, $K_5 = -10$, $K_6 = 0$, $K_7 = 0$, $K_{clutter} = 1$ are user-defined parameters, h'_{BS} and h'_{UE} are the effective BS and UE heights, respectively, by taking into account the earth terrain. L_{diff} is the diffraction loss calculated by Deygout method and $f(clutter)$ is the weighted average of the user-specified clutter losses, in the propagation path between BS and UE [7].

3.4 ITU 452 Model

It is based on the ITU-R P.452-15 recommendation [8] and is valid for 100-500,000 MHz band. It takes into account the LoS/NLoS state, diffraction, tropospheric scatter, surface ducting and elevated layer reflection and refraction. It is given by the following formula:

$$L_{path} = -5 * \log\left(10^{-0.2*L_a} + 10^{-0.2*(L_b+(L_c-L_d)*F_j)}\right) + A_{BS} + A_{UE}, \quad (12)$$

where

$$F_j = 1 - 0.5 * \left[1 + \tanh\left(2.4 * \frac{\theta - 0.3}{0.3}\right)\right].$$

Here L_a is the basic transmission loss due to troposcatter, L_b is the minimum basic transmission loss with LoS propagation and over-sea sub-path diffraction, L_c is the basic transmission loss associated with diffraction and LoS or ducting/layer-reflection enhancements, A_{BS} and A_{UE} are additional losses due to BS and UE surroundings, respectively, F_j is the interpolation factor to take into account the path angular distance and θ is the path angular distance. These parameters are further calculated from equations in ITU-R recommendation P.452-15 [8].

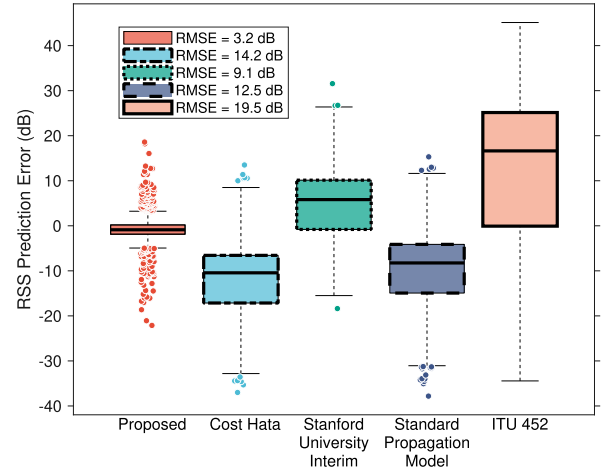


Fig. 6. Comparison of proposed model with various empirical radio propagation models in terms of RSS prediction error, showing a 65% improvement over the best performing empirical model (3.2 dB RMSE as compared to 9.1 dB for SUI).

3.5 Performance Comparison

The proposed ML-based propagation model is compared against traditional empirical propagation models, in terms of predictive performance, generalization performance and computational performance.

3.5.1 Predictive Performance

In Fig. 6, a box-plot representation is used to compare the performance of our proposed model with the state-of-the-art empirical propagation models, by taking highly precise ray-tracing based RSS estimates as ground truth. The data used here as benchmark is unseen and not used earlier in the training or validation process of our proposed ML-based model. The RSS is calculated from the empirical models using $P_{UE} = P_{BS} - L_{path}$, where P_{UE} is the UE's RSS, P_{BS} is the BS's transmit power and L_{path} is the pathloss calculated using (9)–(12). We can see that the predicted RSS using our proposed AI-driven model has much less error as compared to other empirical models, showing a 65% improvement over the best performing empirical model (3.2 dB RMSE as compared to 9.1 dB for SUI).

3.5.2 Generalization Performance

The reason for the gain in accuracy of the proposed model lies in its generalizability as compared to other empirical propagation models. First, ML-based model, thanks to its ability to incorporate higher degrees of freedom compared to an empirical, has an intrinsic advantage over empirical model. Empirical models are usually scenario-specific and have different fine-tuned parameter values for different geographic scenarios (e.g., urban, sub-urban, rural etc.) using extensive channel measurements from that scenario. The key cost of this advantage is opaqueness or black box nature of model, which we will address in the next section.

Second, through the feature engineering process, the proposed model leverages a novel combination of key features, which are not included in traditional empirical models, and can characterize the physical and geometric structure of the environment traversed by a signal in its propagation path (e.g., indoor distance, Manhattan distance, number of

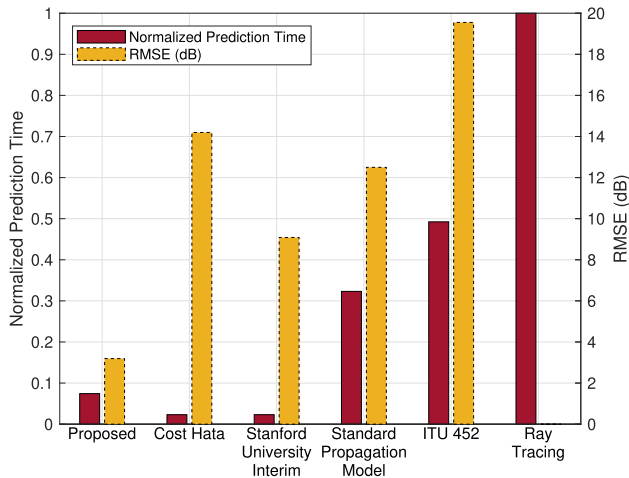


Fig. 7. Comparison of proposed model with various empirical radio propagation models and ray tracing w.r.t. prediction time and RSS prediction error. The proposed ML-based model is atleast 65% more accurate than the investigated empirical models, and 13x faster than ray-tracing.

building penetrations in each clutter type etc.), and are sensitive to the change in network parameters (e.g., horizontal angular separation, vertical angular separation etc.). These additional features (or degrees of freedom) enable the ML-model to be trained on combined data from different geographic scenarios and hence provide more scalability and generalizability.

3.5.3 Computational Performance

While the proposed model yields better accuracy than empirical models, our analysis (shown in Fig. 7) shows that its computational complexity and therefore implementation cost is much lower than the highly sophisticated ray-tracing based tools that are being widely used in commercial cell planning tools, because it only uses the key features as input to the trained ML-based model to predict the RSS, as compared to ray tracing, which approximates the interactions of all rays with the neighboring environment to estimate the pathloss, hence computationally inefficient. As a result, it's much faster than ray-tracing, and thereby addresses a much-complained problem in ray-tracing based tools, by industry professionals. The preliminary implementation of the proposed framework has demonstrated a 13x decrease in prediction time as compared to ray-tracing approach, and can be further optimized to make it more efficient (for instance, by using parallel computing).

4 SECONDARY ANALYSIS FOR INTERPRETABILITY AND SENSITIVITY

One of the key caveats of applying machine learning is lack of interpretability of the resultant models. This challenge often undermines the uptake rate of ML based models, particularly in cellular networks where stakes are high. Therefore, knowing why a model is predicting what it is predicting can be a very useful auxiliary information on top of accuracy, prediction and training agility and robustness to sparsity of training data. Model interpretation is also a vital debugging tool, as it can help you learn about the problems (e.g., biasness) in the model and for ensuring that small changes in the input do

not lead to large changes in the prediction. Therefore, in this section, we try to make our proposed black-box machine learning model more trustable, interpretable and robust [38].

4.1 Sensitivity Analysis

Sensitivity Analysis is a useful technique for investigating the model's behavior for specific scenarios of interest and for providing a global insight into the model's behavior. This is done by quantifying the contribution of each input feature, in the variability of the model output. These values are called *sensitivity indices*.

4.1.1 Sobol Indices

The most popular method of finding these sensitivity indices is *Sobol Method* [39], which is based on the variance of model output. However, they are very difficult to interpret if there is a statistical dependence between features. For instance, in the case of independent features, there exists a unique *Sobol Index* for a feature, representing the variance in model output solely by that feature, also called *First Order Sobol Index*. But if the features have dependency between them, then the first order indices fail to capture the contribution of each feature, and *Second Order Sobol Indices* are used to express the contributions of the interactions between each pair of features, and so on for higher orders.

4.1.2 LOCO Variable Importance

Leave-one-covariate-out (LOCO), or even Leave-one-feature-out (LOFO) [40] is another method for finding variable importance (or sensitivity) in the model output. It scores each row in the training data for each feature (or covariate). In each scoring run, one feature is missed and its impact on the output prediction is measured. The feature with the most impact on the predicted outcome is taken to be the most important. However, its performance can quickly deteriorate if there are complex non-linear dependencies in a model, in which case Shapley values will be a better technique.

4.1.3 Shapley Values

The lack of accurate model interpretation using the above methods, when there are complex non-linear interdependencies between features, can be overcome by using *Shapley values* [41], which is a Nobel-laureate concept in cooperative game theory and economics, to determine the contribution of each player in a collaborative game to its success, but can be used to calculate feature importance in a model and thus achieve a good degree of interpretability, even for non-parametric models. *In the case of dependence between a group of input features, the effect of interaction between features is equally allocated to each feature within the group.*

4.2 Model Interpretation With SHAP

A recently introduced method called *SHAP (SHapley Additive exPlanations)* [42], based on Shapley values, measures how much each feature contributes, either positively or negatively, to the model output. An advantage of using SHAP is that each sample in the data has its own set of SHAP values, unlike traditional methods, which only tells the importance of a feature across the whole dataset. This is particularly

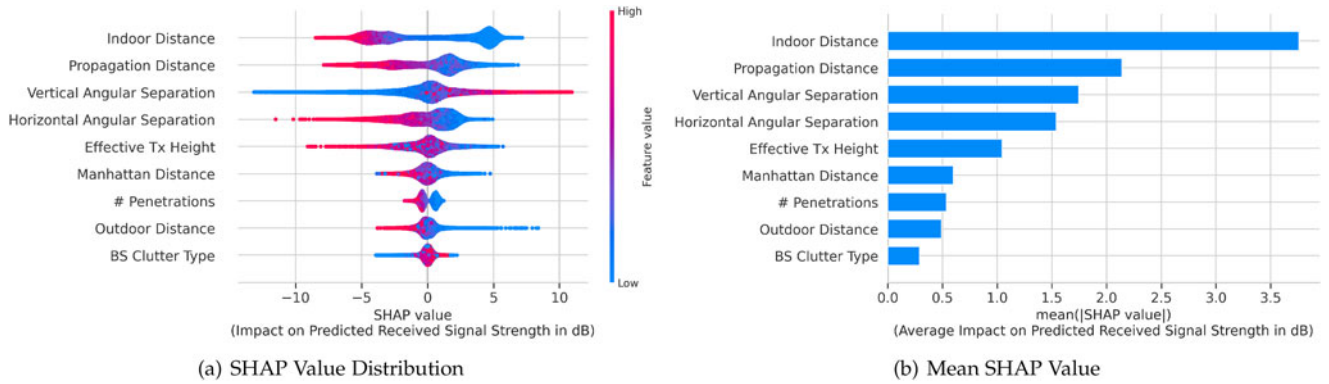


Fig. 8. (a) SHAP summary plots showing top 9 input features (a) SHAP value (Impact on RSS) variance w.r.t respective feature values (b) Mean SHAP values (Average impact on RSS).

useful, as we can observe the effect on model output, for the whole range of each input feature. In our further analysis, we have used the TreeSHAP algorithm [43], which is an efficient approach of calculating shapley values of ML models belonging to decision tree family (e.g., LightGBM, XGBoost etc.).

4.2.1 Feature Importance Using SHAP Summary Plots

In Fig. 8a, SHAP values of some features are plotted for all measurement instances, which show the distribution of impacts on the predicted RSS value, for each input feature. Here the points are colored by the respective feature's value and piled up vertically to show density. For each measurement sample, the sum of SHAP values (for every feature) equals the variance in the predicted output from its mean value across all samples. For instance, from domain knowledge we know that the RSS of a user will be highest if its horizontal angular separation from the BS antenna is close to zero and vice versa, due to the impact of antenna beamwidth on its attenuation. Similarly, the RSS of a user will be high if it's close to the BS, therefore, the impact of propagation distance is highest at its extreme values, same is the case for indoor distance and outdoor distance. On the other hand, Vertical Angular Separation is generally inversely proportional to the distance between BS and UE, therefore its impact is highest when it has a high value and vice versa. Also the impact of Effective BS Height is high, if the net vertical distance between the UE and BS is high, and vice versa. Similar trends can be seen for other features as well.

In Fig. 8b, the mean SHAP value for each input feature is plotted. These results show the average impact of each feature on the model output (i.e., predicted RSS value). We can see that, contrary to the common understanding where distance is considered and used in literature as the key determining factor for pathloss (or RSS), the Indoor Distance has the highest feature importance (or impact) in the RSS model. On the other hand, BS Clutter Type in the propagation path has the lowest impact.

4.2.2 Feature Inter-Dependency Using SHAP Dependency Plots

The interplay of combinations of features can be uncovered using SHAP dependence plots. By plotting the SHAP value for many samples in the dataset (See Fig. 9), we can see that the SHAP value (attributed importance) of a feature

changes as its value varies. However, its interaction with other features in the model is captured by its vertical dispersion. Unlike standard partial dependence plots, that only plot a line, here each dot (sample) is colored with the value of an interacting feature.

In Fig. 9a, we can see that the impact of propagation distance decreases as its value increases. Whereas, as mentioned before, indoor distance has an interaction with the propagation distance that affects its relative importance. Fig. 9b shows the effect of effective BS height on the attributed feature importance of vertical angular separation, where high value of effective BS height decreases the importance of vertical angular separation when its value is greater than zero, and vice versa. Similarly, in Fig. 9c, we can see the increase in feature importance of indoor distance at points where vertical angular separation is high. Fig. 9d shows that feature importance of Manhattan distance and its interplay with horizontal angular separation. In Fig. 9e, as we know that the increase in number of building penetrations in the propagation path between a BS and UE, increase its indoor distance as well in most cases, results in the decrease of its feature importance (Fig. 9e). Lastly, Fig. 9f shows the feature importance of effective BS height and its interaction with vertical angular separation.

4.3 Insights From Interpretability/Sensitivity Analysis

To interpret the model predictions and gain insights into the black-box model by turning it into rather grey-box model, SHAP algorithm is used. The SHAP Summary plot (or the feature importance plot) in Fig. 8 shows the mean importance of each feature in the variability of the model output. This plot is particularly useful for a system-level control as it shows that what control knob (or network parameter) needs to be played the most for tuning network configuration to get optimal performance. SHAP Dependency plots (shown in Fig. 9), on the other hand, shows the behavior of feature importance (or SHAP value) with respect to the value of its corresponding feature and its interaction with the most dependent feature. This plot is useful for observing the range of values for a pair of features that has the highest impact on the model output. For Example, Fig. 9c shows that the indoor distance and vertical angular separation have the highest impact on the model output when $0 < d_{indoor} < 20$ m and $\phi_{ver} > 10^\circ$.

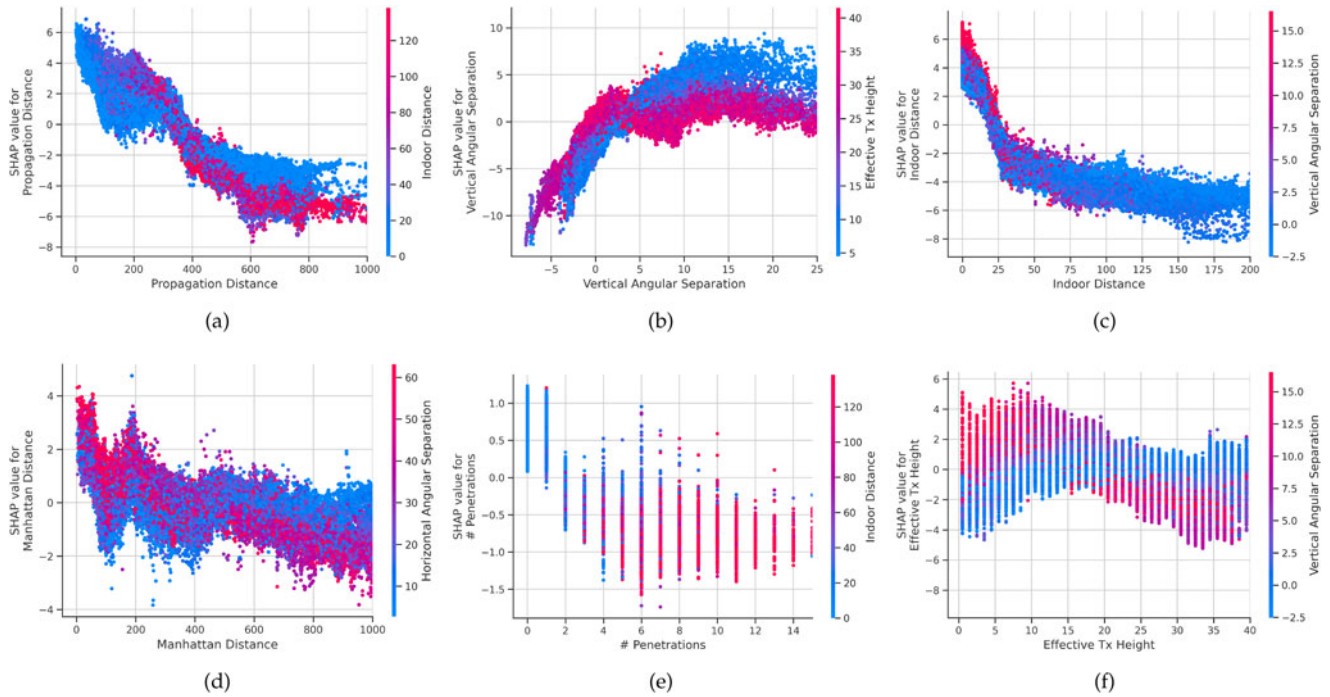


Fig. 9. Impact of various features and their inter-dependence in predicting received signal strength. Useful for model interpretability and finding important feature regions (or values) for intelligent data collection and network automation.

4.4 Utility of Insights Gained From the Proposed Model

The information yield by the SHAP analysis, that has transformed the originally black-box model into a *grey-box model*, can be exploited in real networks for several use cases. Below we identify three key use cases:

- 1) **Addressing the Sparsity Challenge:** A key challenge in applying ML to wireless networks is sparsity of training data i.e., gathering data for complete parameters ranges is often very difficult, if not impossible. For example, its not viable to gather RSS measurements against all antenna tilt range (0-90) in a live network. Furthermore, usually the process of gathering and enriching training data is costly. The proposed framework builds a grey-box model instead of a black box model, thanks to the insights provided by the SHAP analysis, can be leveraged to address the aforementioned challenges of data sparsity. The knowledge that what parameter ranges are more crucial to the model can be exploited for selective collection and enrichment of training data. This can provide a lower cost/benefit ratio as compared to a uniform or random collection or enrichment of training data. For example, based on observation from Fig. 9c, instead of uniform or random measurement campaigns, more resources should be dedicated to data collection for Antenna Tilt and UE's RSS data pairs corresponding to $\phi_{ver} > 10^\circ$ (vertical angular separation) and Antenna Tilt/Azimuth and UE's RSS data pairs corresponding to $0 < d_{indoor} < 20$ (indoor distance in the propagation path).
- 2) **Intelligent Optimization:** Current design and post deployment optimization paradigm of cellular networks rely mostly on the domain knowledge.

However, given the large number of design and optimization parameters per site—already roughly 1500/site in LTE — and growing complexity trend towards 5G and beyond, achieving optimal design and operation in emerging cellular networks by solely relying on domain knowledge is going to become inviable approach. The insights gained from the semi transparency (vis-a-vis greyness) of the presented model achieved through the proposed framework can be very helpful towards more effective and resource efficient design and post deployment optimization of the network. For example, while searching for optimal design and configuration parameters, the parameters and regions of the search space with parameter ranges identified by the proposed framework to be more influential on the KPIs, can be explored more exhaustively, compared to other parameters and parts of the parameter range. This approach is expected to improve the design and optimization processes compared to uniform (brute force based) or pseudo random or heuristic search algorithms (e.g., genetic algorithms, simulated annealing) based design and optimization.

- 3) **Lighter ML model for low-latency use-cases:** The insights gained from the SHAP analysis can also be used to select the most important features for building our proposed ML model. Therefore, a lighter version of the model can be built using the selected key features to further reduce the computational complexity of the model. The results from this analysis (in Table 6) show that by using the top 5 features (from Fig. 8b), i.e., indoor distance, propagation distance, vertical angular separation, horizontal angular separation and effective BS height, the training and prediction time of the resulting RSS prediction model can be

TABLE 6
Performance Evaluation of the SHAP-Enabled Lighter Model Using 5-Fold Repeated Cross Validation

Performance Metric	Baseline ML Model using all features	Lighter ML Model using Top 5 features	Net Gain
RMSE	3.542 ± 0.036	3.661 ± 0.038	-3.35%
R^2	0.854 ± 0.003	0.844 ± 0.003	-1.17%
Training Time	1.896 ± 0.090	0.681 ± 0.049	+64.07%
Prediction Time	0.077 ± 0.005	0.023 ± 0.002	+70.13%

significantly reduced (by around 70%) at the cost of negligible loss in accuracy (by around 3%) to enable low latency use-cases for the proposed SHAP-enabled lighter model. By allowing real-time predictions, such lightweight model can be used for real time AI-powered closed loop optimization of the network, thus acting as a key enabler for the Ultra-Reliable Low-Latency Communication (URLLC) in 5G networks.

5 CONCLUSION AND FUTURE WORKS

In this paper, we propose a framework for a robust and scalable AI-driven 3D propagation model for cellular networks. To enable this framework, we have identified a novel set of key predictors, that can characterize the complex physical and geometric structure of the propagation environment. Performance comparison of several state-of-the-art machine learning algorithms including Linear Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Extremely Randomized Trees, Adaptive Boosting (AdaBoost), Gradient Boosting Decision Trees (GBDT), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting (LightGBM), Categorical Boosting (CatBoost) and Deep Neural network (DNN) is done to highlight their strengths and weaknesses in modeling the propagation through complex real environment using the proposed key predictors as input features. The results show that *LightGBM* outperforms other ML tools, including DNN, in terms of computational complexity and robustness to extremely sparse training data (just 2%), as often is the case in real networks. On the other hand, compared to other tested ML tools, DNN's performance deteriorates the most when the training data becomes sparse. The proposed ML-Based model is compared against state-of-the-art empirical models including COST-Hata, Stanford University Interim (SUI), Standard Propagation Model (SPM) and ITU 452 Model. Proposed ML-based model yields 65% higher accuracy in RSS estimation as compared to empirical propagation models, when highly sophisticated ray-tracing based data for the city of Belgium from a commercial planning tool is used as ground truth. On the other hand, proposed model offers 13x reduction in prediction time as compared to ray-tracing based commercial planning tool. The black box nature of the proposed model is transformed into relatively more interpretable grey-box model using SHAP method. The insights presented through interpretability analysis offer new research directions such as intelligent data gathering for addressing the challenge of training data sparsity, finding the optimal combination of network configuration parameters and building lighter ML models for low-latency use-cases.

The presented system level pathloss prediction model combined with the interpretability analysis thus manifests a framework that can act as a corner stone for the Artificial Intelligence driven Self Organizing Networks (AISON), as opposed to current SON which lacks interpretable models for quantifying network performance as function of the plethora of network configuration parameters. Furthermore, in addition to system level pathloss model, the proposed framework can be extended to an AI-driven link level channel model for channel estimation, physical layer design etc. Such extension will require incorporating many other channel parameters such as delay spreads and angular spreads etc., that can be neglected in system level path loss model due to the much larger temporal and spatial scale that can suffice for system level planning and optimization, but cannot give meaningful insights for link level design. Other possible extensions are developing both link level and system level channel models for higher frequency bands. This is needed especially for next-generation 3D heterogeneous cellular networks in mmWave/Terahertz bands with flying Unmanned Aerial Vehicle (UAV)-based BSs [44], [45] and Reconfigurable Intelligent Surfaces (RIS) [46], since propagation conditions will be significantly different than sub-6 GHz bands and conventional network planning using empirical model will cease to be a viable option.

ACKNOWLEDGMENTS

A portion of this work has been published in IEEE GlobeCom'19 [1].

REFERENCES

- [1] U. Masood, H. Farooq, and A. Imran, "A machine learning based 3D propagation model for intelligent future cellular networks," in *Proc. IEEE Global Commun. Conf.*, 2019, pp. 1-6.
- [2] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: How to empower SON with Big Data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27-33, Nov./Dec. 2014.
- [3] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas Propag. Mag.*, vol. 45, no. 3, pp. 51-82, Jun. 2003.
- [4] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 1, pp. 255-270, First Quarter 2013.
- [5] COST 231, "Urban transmission loss models for mobile radio in the 900 & 1800MHz band," *COST 231 TD (90)*, 119 Rev 2, The Hague, Netherlands, 1991.
- [6] V. Erceg *et al.*, "Channel models for fixed wireless applications," *IEEE 802.16 Broadband Wireless Work. Group*, Jul. 2001.
- [7] Forsk, Model Calibration Guide, AT340_MCG_E0, France, 2018. [Online]. Available: <http://www.forsk.com>
- [8] ITU-R Recommendation P.452-15, "Prediction procedure for the evaluation of interference between stations on the surface of the earth at frequencies above about 0.1 GHz," ITU, Geneva, 2013.
- [9] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2735-2747, Jul. 2010.

- [10] M. Piacentini and F. Rinaldi, "Path loss prediction in urban environment using learning machines and dimensionality reduction techniques," *Comput. Manage. Sci.*, vol. 8, no. 4, pp. 371–385, 2011.
- [11] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on machine learning: Principle, method, and data expansion," *Appl. Sci.*, vol. 9, no. 9, 2019, Art. no. 1908.
- [12] M. Ribero, R. W. Heath, H. Vikalo, D. Chizhik, and R. A. Valenzuela, "Deep learning propagation models over irregular terrain," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 4519–4523.
- [13] S. P. Sotiroidis, S. K. Goudos, K. A. Gotsis, K. Siakavara, and J. N. Sahalos, "Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, no. 1, pp. 364–367, Mar. 2013.
- [14] M. Ayadi, A. B. Zineb, and S. Tabbane, "A UHF path loss model using learning machine for heterogeneous networks," *IEEE Trans. Antennas Propag.*, vol. 65, no. 7, pp. 3675–3683, Jul. 2017.
- [15] S. I. Popoola, E. Adetiba, A. A. Atayero, N. Faruk, and C. T. Calafate, "Optimal model for path loss predictions using feed-forward neural networks," *Cogent Eng.*, vol. 5, no. 1, 2018, Art. no. 144 434 5.
- [16] R. D. Timoteo, D. C. Cunha, and G. D. Cavalcanti, "A proposal for path loss prediction in urban environments using support vector regression," in *Proc. Adv. Int. Conf. Telecommun.*, 2014, pp. 1–5.
- [17] L. Dai, H. Zhang, and Y. Zhuang, "Propagation-model-free coverage evaluation via machine learning for future 5G Networks," in *Proc. IEEE 29th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2018, pp. 1–5.
- [18] A. Ghasemi, "Data-driven prediction of cellular networks coverage: An interpretable machine-learning model," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2018, pp. 604–608.
- [19] Forsk Atoll, 2019. [Online]. Available: <https://www.forsk.com/atoll-overview/>
- [20] Aster Propagation Model, 2019. [Online]. Available: <https://www.forsk.com/aster-propagation-model/>
- [21] Forsk, Aster Technical Reference Guide, AS271_TRG_E0, France, 2019. [Online]. Available: <http://www.forsk.com>
- [22] EGS Technologies, 2019. [Online]. Available: <http://www.egstech.com/>
- [23] H. N. Qureshi and A. Imran, "Optimal bin width for autonomous coverage estimation using MDT reports in the presence of user positioning error," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 716–719, Apr. 2019.
- [24] H. N. Qureshi, A. Imran, and A. Abu-Dayya, "Enhanced MDT-based performance estimation for AI driven optimization in future cellular networks," *IEEE Access*, vol. 8, pp. 161 406–161 426, Sep. 2020.
- [25] A. Karttunen, A. F. Molisch, S. Hur, J. Park, and C. J. Zhang, "Spatially consistent street-by-street path loss model for 28-GHz channels in micro cell urban environments," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7538–7550, Nov. 2017.
- [26] Y. Wang, K. Venugopal, A. F. Molisch, and R. W. Heath, "MmWave vehicle-to-infrastructure communication: Analysis of urban microcellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7086–7100, Aug. 2018.
- [27] M. Krzywinski and N. Altman, "Multiple linear regression," *Nature Methods*, vol. 12, no. 12, pp. 1103–1104, Dec. 2015.
- [28] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [29] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. London, U.K./Boca Raton, FL, USA: Chapman and Hall/CRC, 1984.
- [30] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
- [32] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.
- [33] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.
- [34] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.
- [35] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Berlin, Germany: Springer, 1987, pp. 7–15.
- [36] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, Aug. 1980.
- [37] V. Erceg *et al.*, "An empirically based path loss model for wireless channels in suburban environments," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 7, pp. 1205–1211, Jul. 1999.
- [38] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.
- [39] I. M. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Math. Modelling Comput. Experiments*, vol. 1, no. 4, pp. 407–414, 1993.
- [40] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *J. Amer. Statist. Assoc.*, vol. 113, no. 523, pp. 1094–1111, 2018.
- [41] L. S. Shapley, "A value for N-person games," *Contributions Theory Games*, vol. 2, no. 28, pp. 307–317, 1953.
- [42] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.
- [43] TreeSHAP, 2019. [Online]. Available: <https://shap.readthedocs.io/en/latest/generated/shap.explainers.Tree.html>
- [44] L. Zhang *et al.*, "A survey on 5G millimeter wave communications for UAV-assisted wireless networks," *IEEE Access*, vol. 7, pp. 117 460–117 504, Jul. 2019.
- [45] H. N. Qureshi and A. Imran, "On the tradeoffs between coverage radius, altitude, and beamwidth for practical UAV deployments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2805–2821, Dec. 2019.
- [46] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE Access*, vol. 7, pp. 116 753–116 773, Aug. 2019.



Usama Masood (Graduate Student Member, IEEE) is currently working toward the PhD degree in electrical and computer engineering with the Artificial Intelligence for Networks Center, University of Oklahoma (OU), Tulsa, Oklahoma. His research focus at AI4Networks Center is on designing AI-enabled system-level solutions for Zero Touch Automation in emerging networks. During his PhD, he has contributed to several National Science Foundation (NSF) research and development projects on 5G Cellular Networks including the design

and development of 5G Testbed at OU-Tulsa campus for enabling experimental research on different 5G use-cases and designing advanced cell outage management framework for self-healing networks, a multi-national project on AI-based Self Organizing 5G and beyond networks and a industry collaboration project with Ericsson Research, California for solving the training data sparsity challenge to enable AI-based zero touch automation in emerging networks.



Hasan Farooq (Member, IEEE) received the BSc degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2009, the MSc by research degree in information technology from the Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia, in 2014, and the PhD degree in electrical and computer engineering from the University of Oklahoma, Norman, Oklahoma, in 2018. He has developed several methods to harness unprecedented intelligence from the wireless communication network data that can be

utilized for proactive dynamic network orchestration and optimization. His dissertation was awarded Gallogly College of Engineering Dissertation Excellence Award by University of Oklahoma in 2018. He has contributed to several NSF funded research projects in the domain of self-organizing networks for 5G and beyond and has authored more than 40 high impact publications. He is currently working as AI researcher with Ericsson Research Silicon Valley.



Ali Imran (Senior Member, IEEE) received the BSc degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2005, and the MSc (hons.) degree in mobile and satellite communications and the PhD degree both from the University of Surrey, Guildford, U.K., in 2007 and 2011, respectively. He is currently a presidential associate professor and Willams professor in ECE and founding director of the Artificial Intelligence (AI) for Networks (AI4Networks) Research Center and TurboRAN Testbed for 5G and Beyond

at the University of Oklahoma. His research interests include AI and its applications in wireless networks and healthcare. His work on these topics has resulted in several patents and more than 100 peer-reviewed articles including some of the highly influential papers in the domain of wireless network automation. On these topics, he has led numerous multinational projects, given invited talks/keynotes and tutorials at international forums and advised major public and private stakeholders and co-founded multiple start-ups. He is an associate fellow of the Higher Education Academy, U.K. He is also a member of the Advisory Board to the Special Technical Community on Big Data, the IEEE Computer Society.



Adnan Abu-Dayya (Senior Member, IEEE) received the PhD degree in wireless communications (electrical engineering) from Queens University, Kingston, Canada, in 1992. He has been serving as the founding executive director of the Qatar Mobility Innovations Center (QMIC) since 2009. It is one of the first technology innovation institutions in the Middle East focused on translating R&D and technology innovations into scalable digital platforms and solutions in the field of Intelligent Mobility and Smart Cities. From 2007 to

2008, he served as the chairman with the Electrical Engineering Department, Qatar University. Before moving to Qatar in 2007, he worked for 10 years with AT&T Wireless in Seattle, USA where he served in a number of senior management positions covering product innovations, emerging technologies, systems engineering, product realization, and intellectual property management. Before that, he worked as a senior manager with Nortel Networks in Ottawa, Canada in the Advanced Technology Group, and as a senior consultant with the Communications Research Centre in Ottawa, Canada. He serves as the chairman of the Advisory Board of the Electrical & Computer Engineering Department, Texas A&M University at Qatar, he is a member of the Steering Committee of the Smart Grid Research Center, Texas A&M University at Qatar, and is a member of the advisory board of the Qatar Car Museum. He has 10 issued patents, and more than 100 referred publications.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**