UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ARTIFICIAL INTELLIGENCE-BASED NETWORK MODELING FOR ENABLING ZERO TOUCH AUTOMATION IN NEXT GENERATION CELLULAR NETWORKS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL AND COMPUTER ENGINEERING

BY

USAMA MASOOD
Norman, Oklahoma
2023

ARTIFICIAL INTELLIGENCE-BASED NETWORK MODELING FOR ENABLING ZERO TOUCH AUTOMATION IN NEXT GENERATION CELLULAR NETWORKS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Ali Imran, Chair

Dr. James Sluss

Dr. Samuel Cheng

Dr. Timothy Ford

Acknowledgments

This dissertation stands as a testament not only to my personal and academic development but also as a reflection of the immense support, guidance, and wisdom imparted by a constellation of individuals who, in their own unique ways, have contributed to my journey.

First, I express my deepest gratitude to my advisor, Dr. Ali Imran, who provided me with exceptional support and the opportunity to work on projects of profound impact. His steadfast faith in my abilities and his unwavering encouragement have been invaluable in this rigorous journey.

I would like to express my sincere appreciation to my Ph.D. committee including Dr. James Sluss, Dr. Samuel Cheng and Dr. Timothy Ford, who through their insightful feedback, shaped and refined my research direction and methodology.

In addition, I am extremely grateful for the financial support provided by the National Science Foundation (NSF), Qatar National Research Fund (QNRF), Ericsson, T-Mobile, and AT&T Labs in terms of funded research and employment opportunities during my Ph.D. Their support has enriched my research, aligning it with industry-relevant skillsets and ensuring its financial viability, thereby relieving my financial burden.

My time at OU-Tulsa was greatly enhanced by the unwavering support and advice of the OU-Tulsa family. A special mention goes to Dr. Hazem Refai for his helpful career tips and the OU-Tulsa staff - Krista, Denise, Josh, Lee, and Renee - for their steadfast support in administrative and logistical matters, making my journey smoother.

I am thankful for my AI4Networks Research Center colleagues who have not only supported me academically but also personally. Their support, particularly during the tough times, ensured that I never felt alone in this endeavor. They have been a family away from home, and for that I am truly grateful.

I would also like to express my heartfelt gratitude to Dr. Ijaz Naqvi, my MS thesis advisor,

for introducing me to the captivating world of research. His enthusiasm for exploration and discovery was infectious, sparking in me a fervor that I hope this dissertation duly reflects.

No words can encapsulate the depth of my gratitude for my family. To my parents, who instilled in me the importance of perseverance; my wife, who was a beacon of unwavering support; my daughter, the joy of my life; and to my siblings, who have always been there for me. Your collective encouragement and unwavering faith in me have been my bedrock.

To all those mentioned and to those whose contributions may not have been explicitly recognized here, please know that your support and encouragement have left an indelible imprint on my work and on me as a person. I am forever grateful.

Contents

List of Figures

List of Tables

Abstract

The crux of contemporary network design involves mathematical models to describe the quantitative impact of system components on overall performance. However, these models often grapple with trade-offs between accuracy and complexity, and the burgeoning complexity of wireless networks compounds the problem. Artificial Intelligence (AI) and Machine Learning (ML) techniques present promising solutions, albeit with their own challenges, including integration of domain knowledge, sparsity of training data, and network environment dynamism. To address these challenges, the major contribution of this dissertation is to develop novel AI-based network behavior models, which are domain-aware, interpretable, and robust to sparsity and dynamicity.

It first proposes an interpretable AI/ML-based framework to create a large-scale 3D propagation model by leveraging domain knowledge to create a novel set of key predictors (features) that can characterize the physical and geometric structure of the environment traversed by a signal in its propagation path.

Subsequently, the dissertation introduces a novel deep transfer learning framework to create network behavior models when the available network data is extremely sparse and unrepresentative, especially for system-level network behavior modeling. By leveraging network data from similar (source) cells, the proposed novel framework boosts the performance of conventional deep transfer learning by harnessing the strengths of deep neural networks and extreme gradient boosting methods.

However, in a dynamic network environment (with continuously changing user mobility, traffic demand, etc.), selection of source cells with a similar network environment becomes challenging and can hinder the performance of transfer learning. Thus, the dissertation further presents a novel AI-based domain-aware cell similarity prediction framework, designed to identify similar cells in a spatio-temporally dynamic environment to enable transfer learning for cellular network modeling.

In conclusion, the dissertation paves the way for a future of zero-touch automation solutions in network design and optimization, using AI and ML, which could yield immense economic benefits for network operators and better quality of experience for users. It encapsulates novel frameworks for network behavior modeling and paves the way for future research in this field.

CHAPTER 1

Introduction

1.1 Motivation

Mathematical models are at the heart of all phases of network design, describing in quantitative terms the effect that each system component has on the overall performance. Depending on the complexity of the scenario, an accurate mathematical model might not be available. Moreover, even if available, every model is inherently an approximation, and a trade-off exists between the accuracy of the model and its complexity. Accurate models can be too complex to handle, whereas simple models can just be not accurate enough. This is where Artificial Intelligence (AI) and Machine Learning (ML) techniques are being considered as promising viable solutions and have been proven to be very effective for approximating complex functions with hidden features.

AI-based data-driven network model can be used for system level intelligent network planning. These models can also then be used for post-deployment optimization and zero touch automation in cellular networks by orchestrating a plethora of network parameters to maintain optimal multi-faceted network performance in terms of all-important top-level key performance indicators, without much human involvement.

Any system level network model will try to find the mapping between network configuration and optimization parameters (COPs), e.g., antenna parameters, and the resultant key performance indicators (KPIs), e.g., RF quality, user experience, etc. in the observation area. However, due to the increasing complexity in emerging wireless networks, these network behavior models can have an unfathomable parameter space (possible COP-KPI combinations), which means a lot of experiments are required for accurately modeling the network behavior. We leverage a domain knowledge inspired first principle-based

approach to overcome this limitation in creating system level network behavior models. Through domain knowledge we observe that most complex network behavior models build on modeling/predicting low-level KPIs such as RSRP, SNR, etc. This helps us reduce the size of the modeling problem to degree addressable by AI.

However, the adoption and usability of AI/ML based network models is hindered by the following key challenges:

- **Integrating domain knowledge and model interpretability in ML-based network models:** Cellular network modeling is a complex field with many unique characteristics. Domain knowledge about aspects such as the physics of radio propagation, hardware characteristics, user behavior, and regulatory rules is crucial. By integrating this domain knowledge into machine learning models, they can be guided to focus on key features relevant to the task, thus improving their effectiveness and efficiency. Model interpretability is particularly important in cellular networks as decisions made by these models can have significant impact, such as allocation of bandwidth, prediction of network failures, or managing handovers between cells. Therefore, network operators need to understand how the model is making these decisions. By improving interpretability, operators can gain confidence in the model's decisions, troubleshoot any issues more effectively, and meet any regulatory obligations.

- **Sparsity and uneven distribution (unrepresentativeness) of real network (training) data collected by the operators:** In the context of cellular network modeling, sparsity refers to the lack of sufficient data across different network conditions, like cell tower configurations, geographic locations, user behaviors, etc. Machine learning models need a diverse set of data to generalize well, and a sparse dataset can lead to poor performance. Uneven distribution is also a significant issue in cellular networks. For instance, operators might have extensive data for urban environments with high user density, but limited data for rural or underserved areas. This dis-

parity can lead to models that work well for heavily represented areas but perform poorly in underrepresented ones.

- **Dynamicity of the underlying network environment:** Cellular networks are not static; they change over time and respond dynamically to a variety of factors. These can include changes in network traffic, weather conditions, hardware failures, changes in network configuration, and many others. This dynamic nature of the network environment can make it difficult to model, as the model must be able to adapt to these changes and still make accurate predictions. For example, a model trained on data from the morning might not perform well in the evening when network traffic patterns are likely to be very different.

1.2 Research Objectives

In light of the discussion in section 1.1, this dissertation explores the following research questions:

- Radio Propagation Model to characterize the received signal strength of a user is one of the fundamental example of network behavior model. It is used in almost every aspect of network planning, design and optimization. How can we leverage domain knowledge to engineer features for an AI/ML-based propagation model that can better characterize the environment traversed by a signal in its propagation path?
- ML techniques are suitable for modeling the complex network propagation environment in real-time, especially in terms of their generalization ability, computational cost, and predictive power?
- A key caveat of using machine learning (ML) is the lack of interpretability of resultant models, i.e., the black box paradox. How can we make ML models interpretable and is there any utility of the insights gained from the interpretable models in enabling automation?

- How to improve the performance of deep transfer learning in sparse data scenarios as deep learning models require a lot of training data, even for transfer learning (fine-tuning). Can we make advances in machine learning to improve the performance of conventional deep transfer learning?

- How to efficiently tune the hyperparameters of deep transfer learning model based on the available data in target domain (sparsity of network experiments in target cell)?

- How to quantify the effects of spatial dynamicity (of different geographical environments e.g., urban, suburban, rural, etc.) in the network environment for creating a cell similarity prediction framework for transfer learning?

- How to quantify the effects of temporal dynamicity (e.g., changing user mobility and traffic pattern) in the network environment while creating a cell similarity prediction framework for transfer learning?

1.3 Contributions

The primary contributions of this dissertation are summarized as follows:

- We present a framework for an Artificial Intelligence (AI)-driven large-scale 3D pathloss model that is scalable and robust to the variations in the environment geography and addresses the limitations of conventional pathloss modeling.

- 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*).

- 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.

- 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%.

- 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.

- A key caveat of using Machine Learning (ML) is the lack of interpretability of resultant models, i.e., the black box paradox. In this dissertation, we try to address this weakness of the proposed Received Signal Strength (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.

- Another key contribution of the dissertation 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 has $\sim 70\%$ less computational complexity compared to the base line model at the cost of negligible loss in performance (around 3%).

- This dissertation presents a joint modeling approach for cellular network behavior, specifically focusing on cell-level Key Performance Indicators (KPIs) such as Avg. RSRP, Avg. RSRQ, and Avg. SINR. These indicators are examined in the context of variable antenna configuration parameters like antenna tilt angle, maximum transmit power, and antenna height. The goal is to better understand and optimize network planning. The training data required for this system-level modeling is sourced through numerous network experiments involving the modification of antenna parameters and subsequent analysis of their impact on KPIs.

- A novel method for transfer learning is proposed, utilizing a boosting cascade ensemble of Deep Neural Networks (DNN) and Extreme Gradient Boosting models (XGBoost). Compared to traditional DNN-based transfer learning methods, the proposed model is projected to improve models' prediction accuracy by 25% given the same volume of training data. This increase in efficiency also means fewer network experiments are necessary to achieve the same performance, potentially reducing costs for mobile network operators.

- Extensive transfer learning experiments are conducted, analyzing different transfer learning strategies across a variety of data sparsity conditions. These experiments provide insights that are instrumental in designing an optimal transfer learning

model based on the data available in the target domain. Such insights are particularly useful in situations where ground truth data is sparse or unavailable.

- A robust and scalable domain-knowledge-based similarity metric is proposed, capable of handling differences in geography, user traffic, and user mobility patterns. This 44-dimensional ML-based similarity metric (SM) includes latent features unaffected by data changes, such as eNB configuration and optimization parameters (COPs), facilitating the identification of similarities between cells.

- The proposed similarity metric is compared with current distance and divergence metrics, and the relative improvements from the proposed transfer learning methodology over the existing state-of-the-art method are quantified. The proposed ML-based similarity metric (SM) shows less than 6% Normalized Root Mean Square Error (NRMSE), suggesting it can predict the gains achievable through transfer learning between two eNBs with a normalized accuracy of $\pm 6\%$, even prior to potential network experiments.

- Ultimately, a decision tree diagram is proposed to aid in selecting the optimal deep transfer learning strategy, considering both the data sparsity in the target domain and the similarity between cells. This diagram provides a practical tool to guide strategy selection in the complex landscape of deep transfer learning applications in cellular network modeling.

1.4 Dissemination and Publications

During the preparation of this dissertation, various dissemination activities were conducted. These efforts have led to several presentations and peer-reviewed articles, which are either accepted or under review.

Patents

P1. U. Masood, and Ali Imran, "Novel ML-based Domain Aware Cell Similarity Prediction Framework to enable Advanced Transfer Learning in a Dynamic and Sparse network" (application in process).

Journal articles

J1. U. Masood, H. Farooq, A. Imran, and A. Abu-Dayya, "Interpretable AI-based Large-scale 3D Pathloss Prediction Model for enabling Emerging Self-Driving Networks," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2022.3147191.

J2. H. N. Qureshi, U. Masood, M. Manalastas, S. M. A. Zaidi, H. Farooq, J. Forgeat, M. Bouton, S. Bothe, P. Karlsson, A. Rizwan, and A. Imran, "Towards Addressing Training Data Scarcity Challenge in Emerging Radio Access Networks: A Survey and Framework," in IEEE Communications Surveys and Tutorials, doi: 10.1109/COMST.2023.3271419.

J3. M. S. Riaz, H. Qureshi, U. Masood, A. Rizwan, A. Abu-Dayya, and A. Imran, "A Hybrid Deep Learning-based (HYDRA) Framework for Multi-Fault Diagnosis using Sparse MDT Reports," in IEEE Access, doi: 10.1109/ACCESS.2022.3185639.

J4. M. C. Manalastas, M. Nabeel, A. Ijaz, S. M. A. Zaidi, U. Masood, H. N. Qureshi, H. Refai, and A. Imran, "Design Considerations and Deployment Challenges for Turbo-RAN 5G and Beyond Testbed," in IEEE Access, vol. 10, pp. 39810-39824, 2022, doi: 10.1109/ACCESS.2022.3166947.

J5. H. Qureshi, U. Masood, A. Abu-Dayya, and A. Imran, "Outage Detection for Emerging Networks: Key Challenges and Solutions," IEEE Network. *(under review)*

J6. M. C. Manalastas, M. U. B. Farooq, H. Qureshi, U. Masood, S. M. A. Zaidi, and A. Imran, "Towards Zero-Touch Optimization in Emerging Cellular Networks: A Framework to Address Key Challenges in Leveraging AI," IEEE Communications Magazine. *(under review)*

J7. U. Masood, W. Raza, F. A. Khan, and A. Imran, "Towards Few-Shot Learning: A Novel Deep Transfer Learning Approach for creating Network Behavior Prediction Models using Sparse Data," IEEE Transactions on Network Science and Engineering. *(under internal review)*

J8. U. Masood, W. Raza, F. A. Khan, and A. Imran, "Where to Transfer: AI-based Domain-Aware Similarity Metric to enable Transfer Learning in Emerging Cellular Networks," IEEE Transactions on Cognitive Communications and Networking. *(under internal review)*

J9. U. Hashmi, U. Masood, and A. Imran, "Data-driven Optimization for User-Centric Stienen Networks," IEEE Transactions on Cognitive Communications and Networking. *(under preparation)*

J10. A. Asghar, U. Masood, H. Qureshi, and A. Imran, "An AI-based Coordination Framework for Conflict Avoidance in Automated Network Functions: A Key Step towards Zero-Touch Automation," IEEE Computational Intelligence Magazine. *(under preparation)*

J11. U. Masood, and A. Imran, "Semi Supervised ML-based Spatio-temporal Analysis of Sleeping Cell Detection in mmWave 5G-NR HetNets," IEEE Transactions on Network and Service Management. *(under preparation)*

J12. U. Masood, and A. Imran, "A Distributed Self-Coordinated Framework for Joint Optimization of CCO, MLB and ES in Ultra-Dense mmWave 5G-NR HetNets using Multi-Agent Deep Transfer Reinforcement Learning," IEEE Transactions on Network and Service Management. *(planned)*

Conference Papers

C1. U. Masood, H. Farooq, and A. Imran, "A Machine Learning Based 3D Propagation Model for Intelligent Future Cellular Networks," 2019 IEEE Global Communications

Conference (GLOBECOM).

C2. M. S. Riaz, H. N. Qureshi, U. Masood, A. Rizwan, A. Abu-Dayya and A. Imran, "Deep Learning-based Framework for Multi-Fault Diagnosis in Self-Healing Cellular Networks," 2022 IEEE Wireless Communications and Networking Conference (WCNC).

C3. J. Shodamola, H. Qureshi, U. Masood, and A. Imran, "Towards Addressing the Spatial Sparsity of MDT Reports to Enable Zero Touch Network Automation," 2021 IEEE Global Communications Conference (GLOBECOM).

C4. J. Shodamola, U. Masood, M. Manalastas, and A. Imran, "A Machine Learning based Framework for KPI Maximization in Emerging Networks using Mobility Parameters," 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom).

C5. S. Bothe, U. Masood, H. Farooq, A. Imran, "Neuromorphic AI Empowered Root Cause Analysis of Faults in Emerging Networks," 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom).

1.5 Organization

This dissertation is structured as follows. Chapter 2 of this dissertation proposes an interpretable AI/ML-based framework to create a large-scale 3D propagation model by leveraging domain knowledge to create a novel set of key predictors (features) that can characterize the physical and geometric structure of the environment traversed by a signal in its propagation path. Chapter 3 then proposes a novel transfer learning framework to create network behavior models when the available network data is extremely sparse and unrepresentative, especially for system-level network behavior modeling. By leveraging network data from similar (source) cells, the proposed novel framework boosts the performance of conventional deep transfer learning by harnessing the strengths of deep neural networks and extreme gradient boosting methods. However, in a dynamic network envi-

ronment (with continuously changing user mobility, traffic demand, weather, vegetation, etc.), selection of source cells with similar network environment becomes challenging and can hinder the performance of transfer learning. Chapter 4 then proposes a novel AI-based domain-aware cell similarity prediction framework leveraging a novel set of readily available features that are robust to this concept drift problem and can identify similar cells in a spatio-temporally dynamic environment to enable transfer learning for cellular network modeling. Chapter 5 concludes this dissertation and outlines future research directions to enable zero-touch automation solutions building on the insights from this dissertation and prior works.

CHAPTER 2

Novel Interpretable Machine Learning for Large-Scale 3D Propagation Modeling

2.1 Introduction

Emerging cellular networks are anticipated to witness a dramatic growth in connected devices and exciting new vertical services. AI enabled end to end network automation vis-a-vis next generation Self Organizing Network (AISON), as proposed in [1], 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 [1]. 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., 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 [2, 3]; *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 domi-

nant 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 [4], Stanford University Interim (SUI) [5], Standard Propagation Model (SPM) [6] and ITU-R P.452-15 [7] 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.

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 [1], 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.

2.1.1 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 [8], urban [9, 10] and volcanic ocean islands [11], and seems unable to scale to other environment settings. The authors in [12] 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 [13] 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 [14] 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 [15] 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 [16] 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 [17] 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.

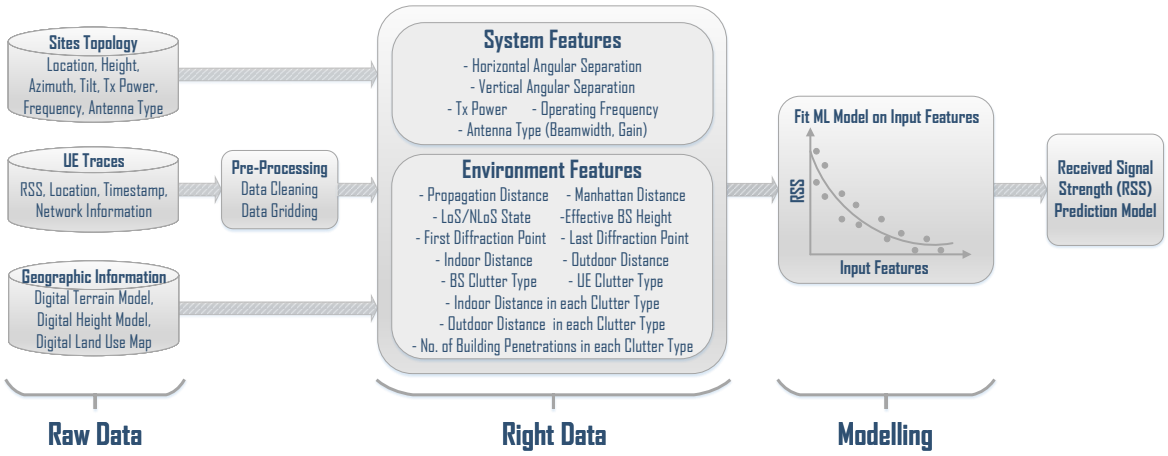


Figure 2.1: Proposed Framework of a Machine Learning-based 3D Radio Propagation Model. Raw datasets such as BS sites topology, UE measurement traces and geographical information are first pre-processed and then converted into right data i-e- feature matrix comprising of key features/predictors that can estimate pathloss (or RSS). These key features are then used to train state-of-the-art ML algorithms for creating an RSS prediction model.

2.2 Proposed Framework

In this chapter, we present a framework for an AI-driven large-scale 3D pathloss model (See Figure 2.1), that is scalable and robust to the variations in the environment geography and addresses the limitations of aforementioned studies.

The proposed framework for an AI-driven 3D propagation model (Figure 2.1) 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 [1], which is then fed to a ML model to estimate RSS at given locations in a radio propagation environment.

2.2.1 Network and Simulation Setup

- Network Scenario: A ray-tracing based industry standard radio network planning and optimization platform "Atoll" [18] is used to create a sophisticated network topology, consisting of 10 macro cell sites in the center of City of Brussels, Belgium (See Figure 2.2(b)). Actual antenna pattern and antenna heights are used in our analysis. Table 3.1 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 [19].

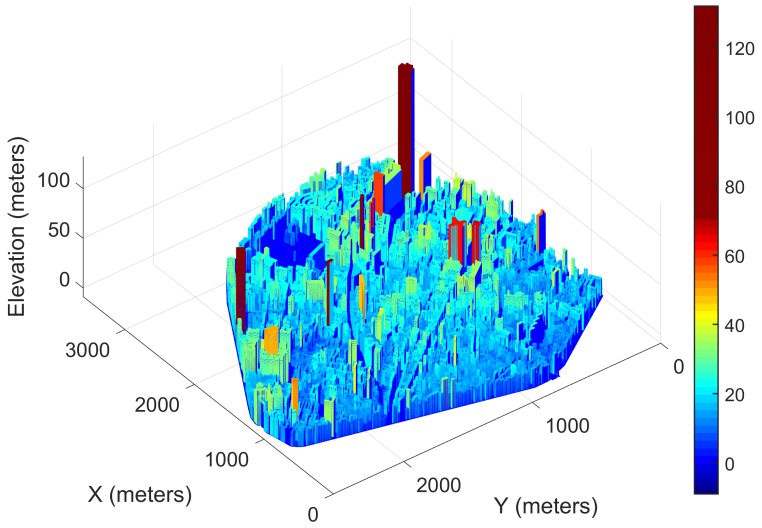
- Ray Tracing: Aster utilizes advanced ray-tracing propagation techniques to calculate various phenomena that affects radio wave propagation including vertical diffractions over roof-tops, horizontal diffractions and reflections based on ray-launching, atmospheric absorptions, rain attenuation and vegetation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

Table 2.1: Network Scenario Settings

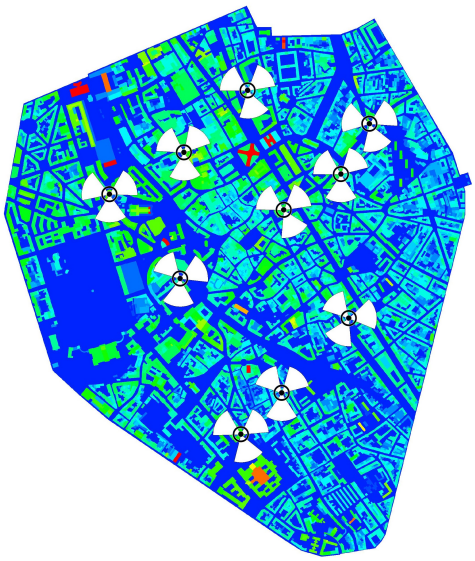
System Parameters	**Values**
Air Technology	4G LTE
Cellular Layout	10 Macrocell sites
Sectors	3 sectors per eNB
Simulation Location	Brussels, Belgium
Simulation Area	3.80 km^2
User Distribution	Poisson Distribution
Propagation Model	Aster (Ray Tracing)
Path Loss Matrix Resolution	10m
Geographic Information	(*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^o
eNB Antenna Vertical Half Power Beamwidth	4.7^o
Frequency Band	2110 FDD (E-UTRA Band 1)
Channel Bandwidth (CBW)	5 MHz

etc. 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.

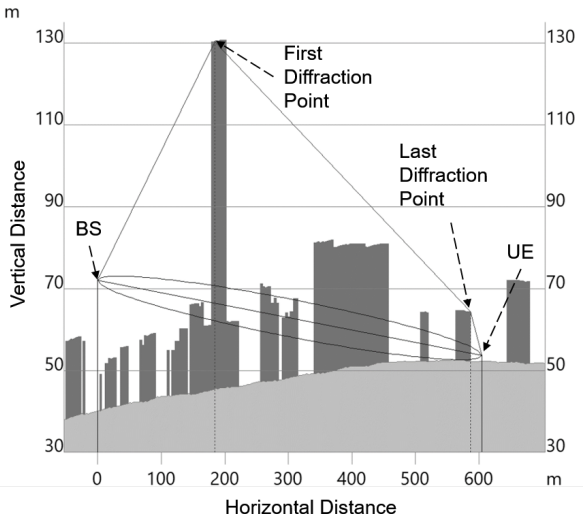
- Validation using Real Data: Furthermore, the parameters of the Aster propagation model are pre-calibrated using more than 1.5 million real channel measurement points from the real environment [20]. Therefore, the high-fidelity RSS (or pathloss) data calculated by Atoll in the observation area can be taken as ground truth for



(a) 3D Elevation Map



(b) 2D Elevation Map



(c) Propagation Path between a BS and UE

Figure 2.2: Area of Simulation showing (**a**) Building Heights (**b**) Transmitter Positions and (**c**) Vertical Propagation Path

designing a realistic propagation model.

2.2.2 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).

Table 2.2: Sites Topology

Parameter	**Description**
Location (x_{BS}, y_{BS})	Location coordinates of a BS site (See Figure 2.2(b))
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

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 2.3). These geographical datasets are routinely used by mobile telecom industry for their planning and maintenance tasks, and can be acquired on demand [21].

Table 2.3: 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 Figure 2.2(c))
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 Figure 2.2(a))
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 $c = D_{DLU}(x, y)$

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 2.4). 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.

Table 2.4: 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.

2.2.3 Data Pre-processing

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

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.

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

2. Handling Positioning Error: Secondly, 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 vs 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 [22] and [23]. Analysis presented in [22] and [23] 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.

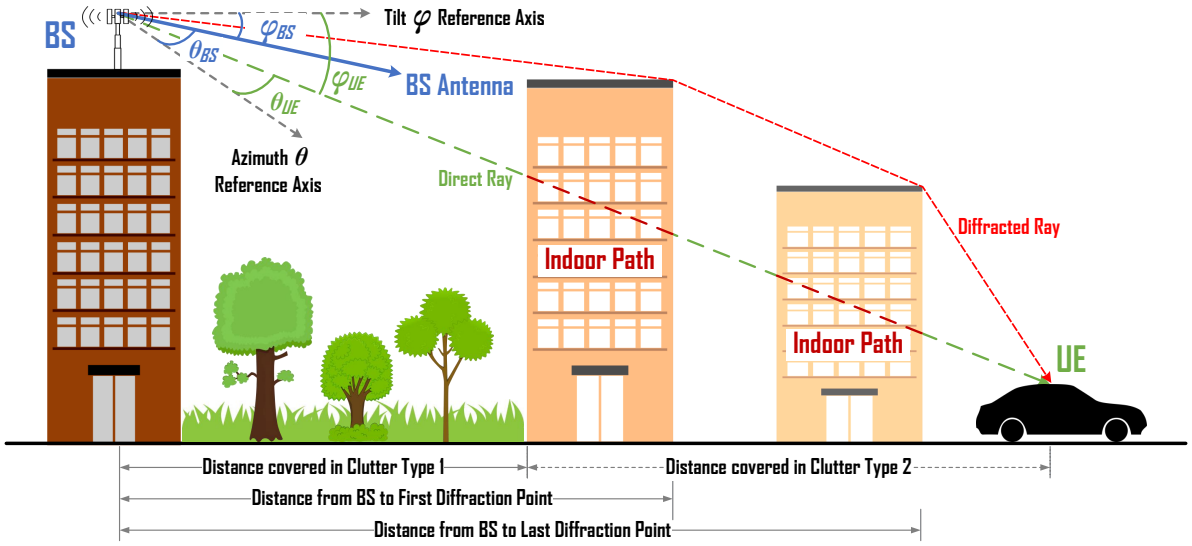


Figure 2.3: Propagation Path between a BS and UE, showing various features of the proposed model

2.2.4 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 char-

acterize the environment traversed by a signal in its propagation path (See Figure 2.3). 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.

Propagation Distance

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

$$d = \sqrt{(x_{BS} - x_{UE})^2 + (y_{BS} - y_{UE})^2}. \tag{2.1}$$

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}.

$$\theta_{hor} = abs(\theta_{BS} - \theta_{UE}), \tag{2.2}$$

where,

$$\theta_{UE} = atan2\left(\frac{x_{BS} - x_{UE}}{y_{BS} - y_{UE}}\right). \tag{2.3}$$

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, $atan2()$ calculates the four quadrant inverse tangent.

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}.

$$\phi_{ver} = \phi_{UE} - \phi_{BS}, \tag{2.4}$$

where,

$$\phi_{UE} = atan\left(\frac{z_{UE} - z_{BS}}{d}\right), \tag{2.5}$$

$$z_{UE} = D_{DTM}(x_{UE}, y_{UE}) + D_{DHM}(x_{UE}, y_{UE}), \tag{2.6}$$

$$z_{BS} = D_{DTM}(x_{BS}, y_{BS}) + D_{DHM}(x_{BS}, y_{BS}). \tag{2.7}$$

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 2.3 for details on D_{DTM} and D_{DHM}).

Effective BS Height

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

$$d_{vert} = z_{BS} - z_{UE}. \tag{2.8}$$

Manhattan Distance

This represents the Manhattan distance between the BS and UE. As radio waves also diffracts at the street corners and 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 [24, 25]. It is denoted by d_{man}.

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.

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}.

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}.

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.

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}.

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 $d_{outdoor}$.

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}.

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}.

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.

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}.

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 $d_{outdoor_c}$.

Table 2.5: Key Symbol Definitions

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

2.2.5 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 chapter, 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 [16] were mostly based on parametric models and some ensemble learning models.

Criteria for Model Evaluation and Selection

In this chapter, 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 chapter 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

Algorithm 1 RSS Prediction Model Training Algorithm

Input: D_{UE}, D_{BS}, D_{DTM}, D_{DHM}, D_{DLU}
Output: RSS Prediction Model $\boldsymbol{\mathcal{M}}(F)$

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

(RMSE) and coefficient of determination (R^2) performance metrics defined below to judge the predictive performance of models.

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

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

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.

Model Evaluation

1. Linear Regression: As evidenced by our experiments, linear regression method [26] doesn't seem to be suitable to capture the complex non-linear nature of the wireless channel. In our results, we have shown (in Figure 2.4) that it gives a high prediction RMSE of *5.45 dB* and a low R^2 score of *0.65*.

2. k-Nearest Neighbors: k-Nearest Neighbors (k-NN) [27] 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 (Figure 2.4).

3. Decision Tree: A single Decision Tree (DT) [28] 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 [29] 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 Figure 2.4), 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 Figure 2.4).

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 Figure 2.4, 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 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 Figure 2.4).

8. Extreme Gradient Boosting: Extreme Gradient Boosting (XGBoost) [30] 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 overfitting and improve overall performance of the model (See Figure 2.4), due to which it is also sometimes called as *regularized gradient boosting*.

9. Light Gradient Boosting: Light Gradient Boosting Machine (LightGBM) [31] 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 Figure 2.4).

10. Categorical Boosting: Categorical Boosting (CatBoost) [32] 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 (Figure 2.4), 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 (Figure 2.4), DNN performs worse than

ensemble-based methods and also has the highest computational cost.

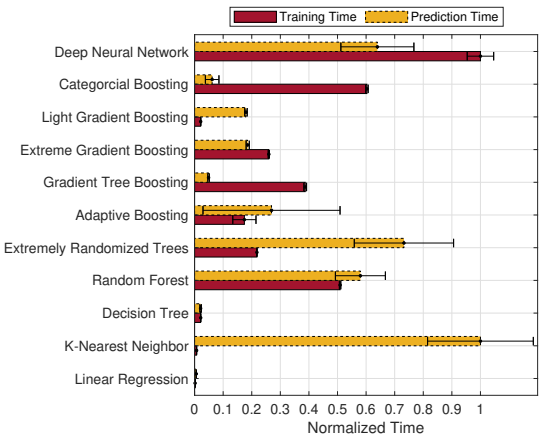
Model Selection

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.

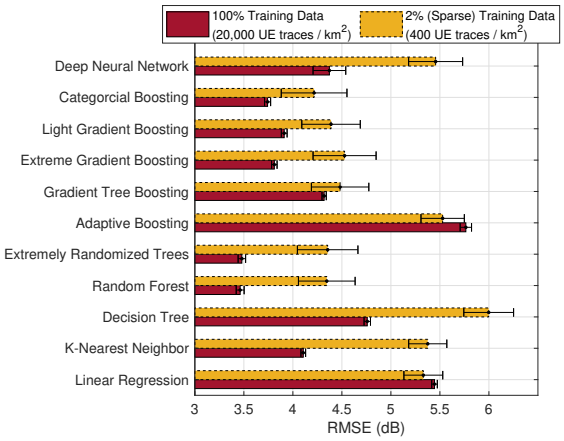
In Figure 2.4(a), training and prediction time of all the methods are normalized to the highest value individually. DNN has the highest, whereas linear regression has the lowest training and prediction time. In Figure 2.4(c), comparison of R^2 value is given, where CatBoost and LightGBM algorithms perform 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 Figure 2.4(b)), 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.

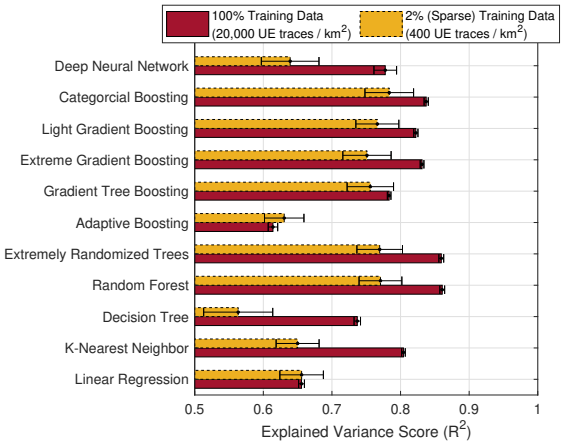
Overall, LightGBM algorithm outperforms others, especially for real-time implementation, due to its lightning fast training process. RSS model trained using LightGBM algorithm is used for further simulations and results. However, it is worth mentioning that the relative superiority of LightGBM model as compared to other algorithms such as Deep Neural Networks etc. might not stay the same if the training data is increased. Therefore, no conclusion should be drawn from this analysis regarding the superiority of an ML algorithm. The above results and insights will hold only for our training data setup and will likely change if the amount of data is changed.



(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

Figure 2.4: 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)

Model Improvement using Hyperparameter Optimization

The baseline LightGBM model performance shown in Figure 2.4 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 Figure 2.5) in terms of performance gain and convergence time:

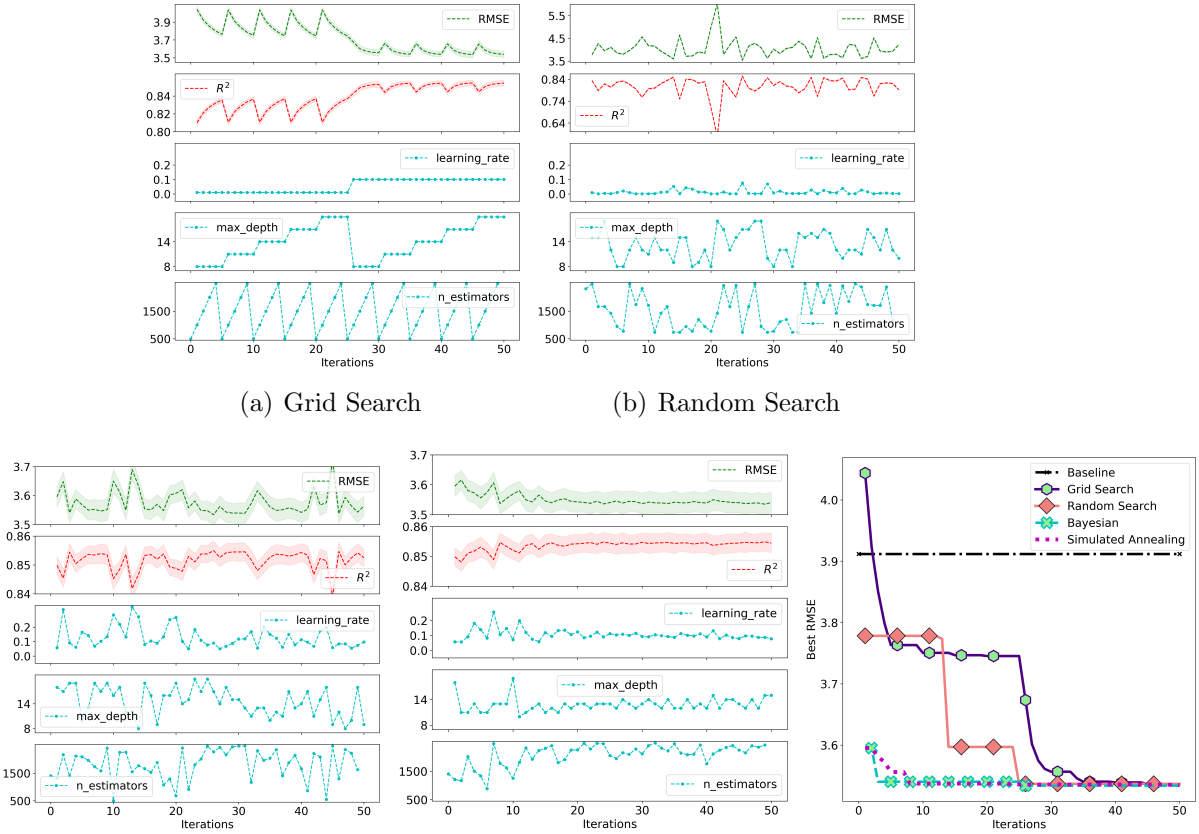
1. Grid Search: This approach does an exhaustive search over the entire search space of hyperparameters. Figure 2.5(a) shows the mean RMSE and R^2 of LightGBM model at different combinations of hyperparameters. The model converges to its best RMSE after 50 iterations (as shown in Figure 2.5(e)).

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 Figure 2.5(e), 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) [33]. 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. Figure 2.5(c) and Figure 2.5(e) 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 [34], that is simpler and is preferred over its Bayesian counterpart when the objective function is simple to evaluate. But it seems to converge slowly than the Bayesian

approach (as evidenced in Figure 2.5(e), 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.



(a) Grid Search

(b) Random Search

(c) Bayesian TPE

(d) Simulated Annealing

(e) Comparison of hyperparameter tuning approaches w.r.t. Performance Gain and Convergence Time. Bayesian optimization performed the best here by achieving a 10% improvement in prediction RMSE as compared to baseline LightGBM model, in just 3 search iterations

Figure 2.5: 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)

2.3 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.

2.3.1 COST-Hata Model

It is an empirical model for pathloss calculation [4], that extends the Hata formulae [35] 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.

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

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:

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

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:

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

2.3.2 Stanford University Interim (SUI) Model

It is derived from the Erceg-Greenstein propagation model [36] and is valid for 1900-6000 MHz. It also takes into account the topo map (DTM). It uses the following formula:

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

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 [5].

2.3.3 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:

$$L_{path} = K_1 + K_2 * log(d) + K_3 * log(h'_{BS}) + K_4 * L_{diff} + K_5 * log(d) * log(h'_{BS}) + K_6 * h'_{UE} + K_7 * log(h'_{UE}) + K_{clutter} * f(clutter). \quad (2.9)$$

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 [6].

2.3.4 ITU 452 Model

It is based on the ITU-R P.452-15 recommendation [7] 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:

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

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 [7].

2.3.5 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.

Predictive Performance

In Figure 2.6, 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 (2.9)-(2.10). 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).

Generalization Performance

The reason for the gain in accuracy of the proposed model lies in its generalizability as compared to other empirical propagation models. Firstly, 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.

Secondly, 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 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.

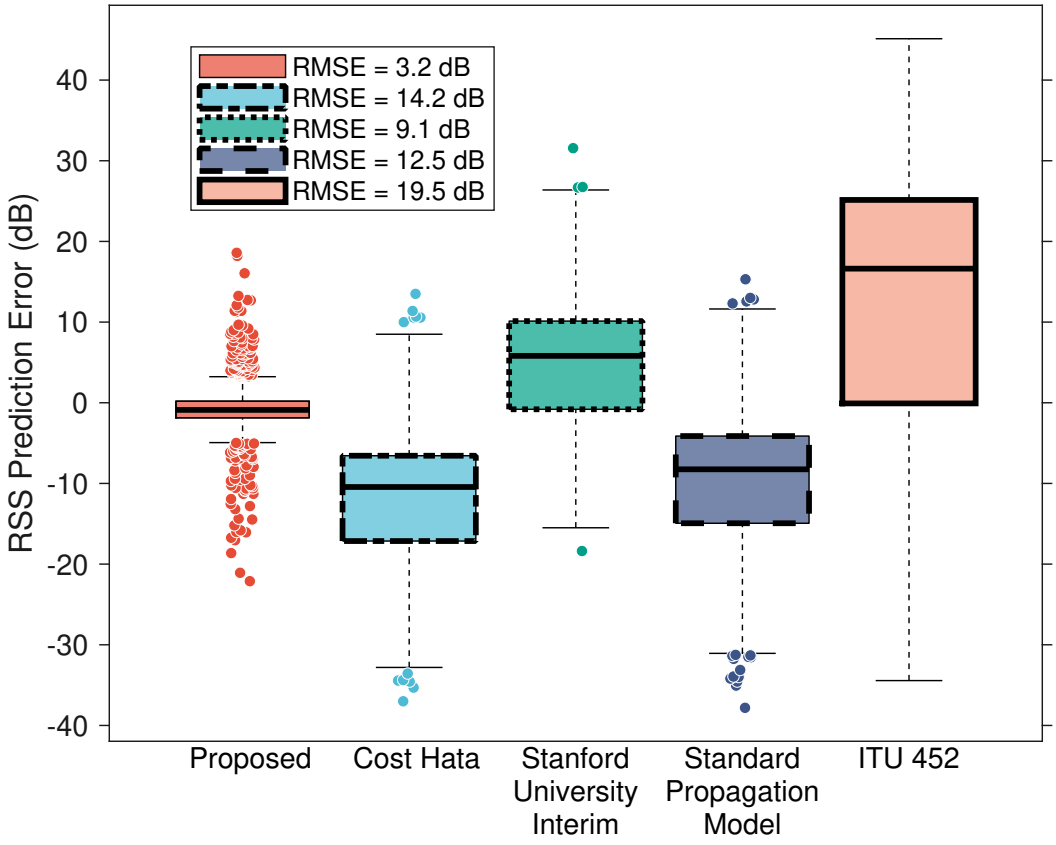


Figure 2.6: Comparison of Proposed model with various Empirical Radio Propagation Models in terms of RSS prediction error, showing a 65% improvement over the best performing empirical model (3.2 dB RMSE as compared to 9.1 dB for SUI)

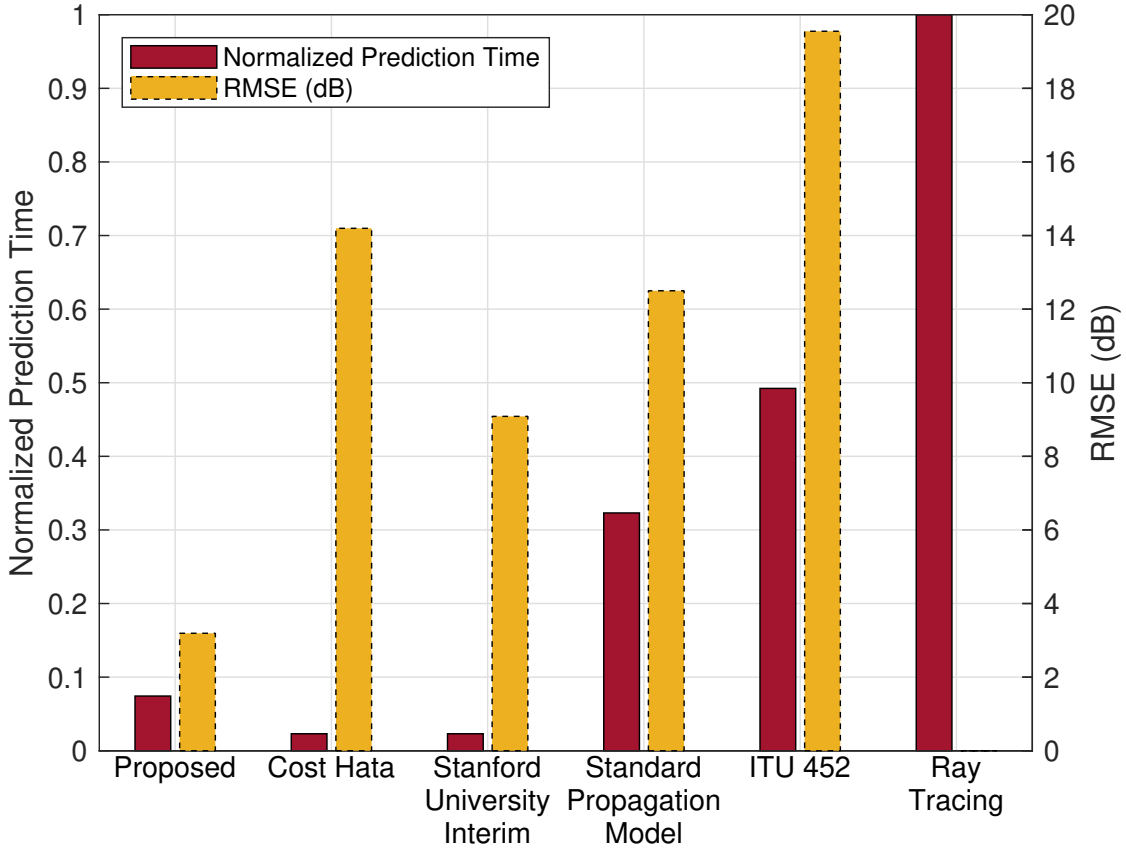


Figure 2.7: Comparison of Proposed model with various Empirical Radio Propagation Models and Ray Tracing w.r.t. Prediction Time and RSS Prediction Error. The proposed ML-based model is atleast 65% more accurate than the investigated empirical models, and 13x faster than ray-tracing

Computational Performance

While the proposed model yields better accuracy than empirical models, our analysis (shown in Figure 2.7) 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).

2.4 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 [37].

2.4.1 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*.

Sobol Indices

The most popular method of finding these sensitivity indices is *Sobol Method* [38], 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.

LOCO Variable Importance

Leave-one-covariate-out (LOCO), or even Leave-one-feature-out (LOFO) [39] 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.

Shapley Values

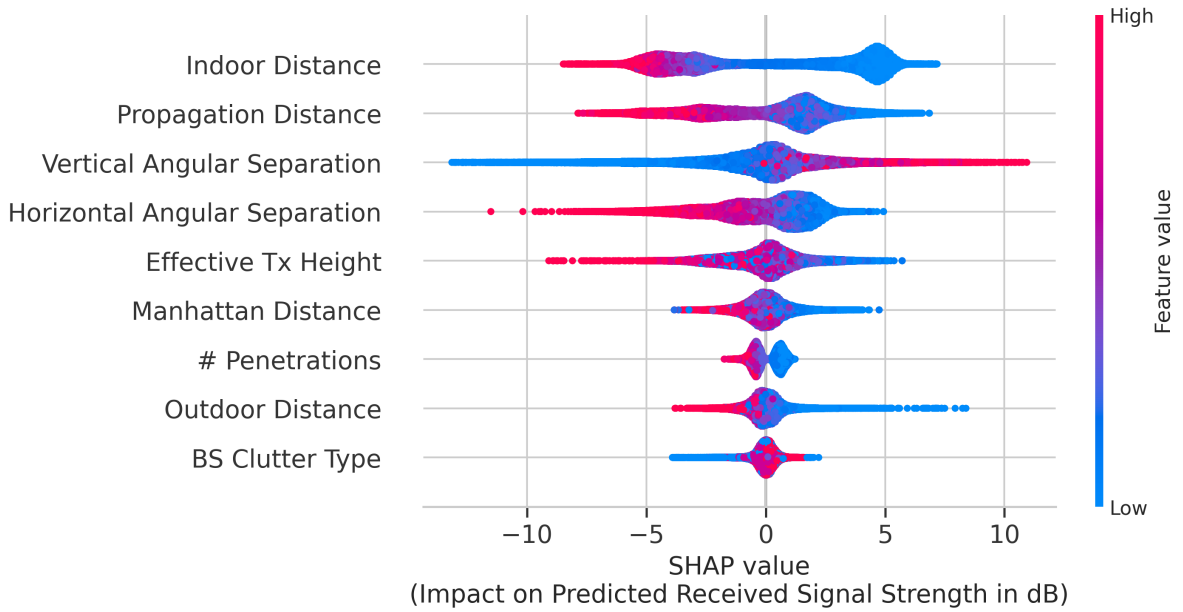
The lack of accurate model interpretation using the above methods, when there are complex non-linear interdependencies between features, can be overcome by using *Shapley values* [40], which is a Nobel-laureate concept in cooperative game theory and economics, to determine the contribution of each player 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.*

2.4.2 Model Interpretation with SHAP

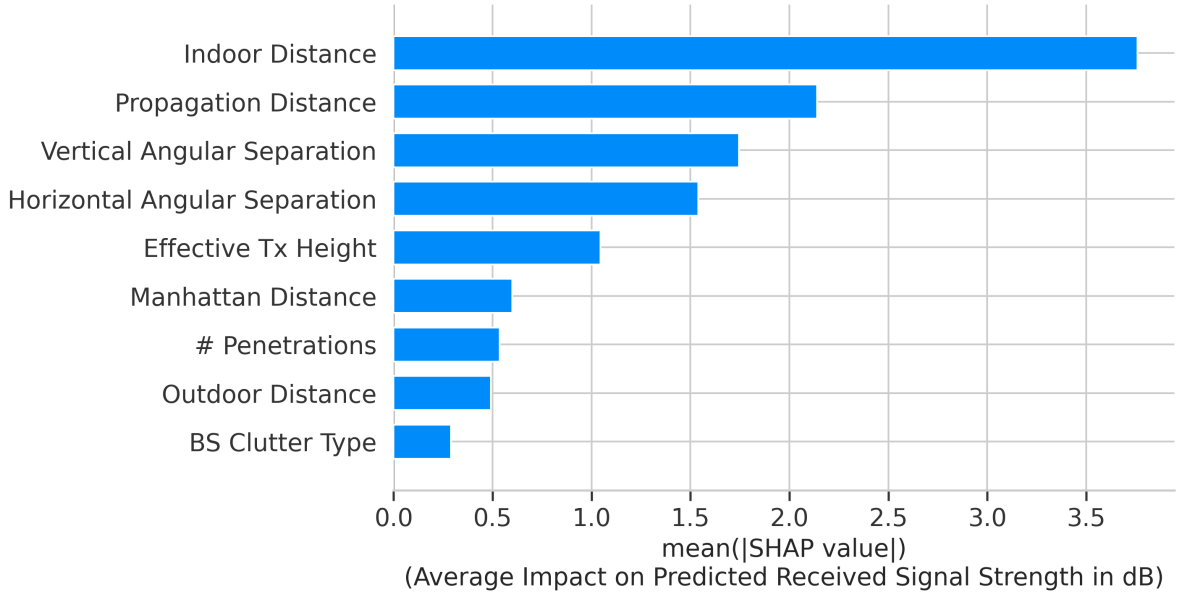
A recently introduced method called *SHAP (SHapley Additive exPlanations)* [41], based on Shapley values, measures how much each feature contributes, either positively or negatively, to the model output. An advantage of using SHAP is that each sample in the data has its own set of SHAP values, unlike traditional methods, which only tells the importance of a feature across the whole dataset. This is particularly 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 [42], which is an efficient approach of calculating shapley values of ML models belonging to decision tree family (e.g., LightGBM, XGBoost etc.).

Feature Importance using SHAP Summary Plots

In Figure 2.8(a), 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.



(a) SHAP Value Distribution



(b) Mean SHAP Value

Figure 2.8: (**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)

In Figure 2.8(b), 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*

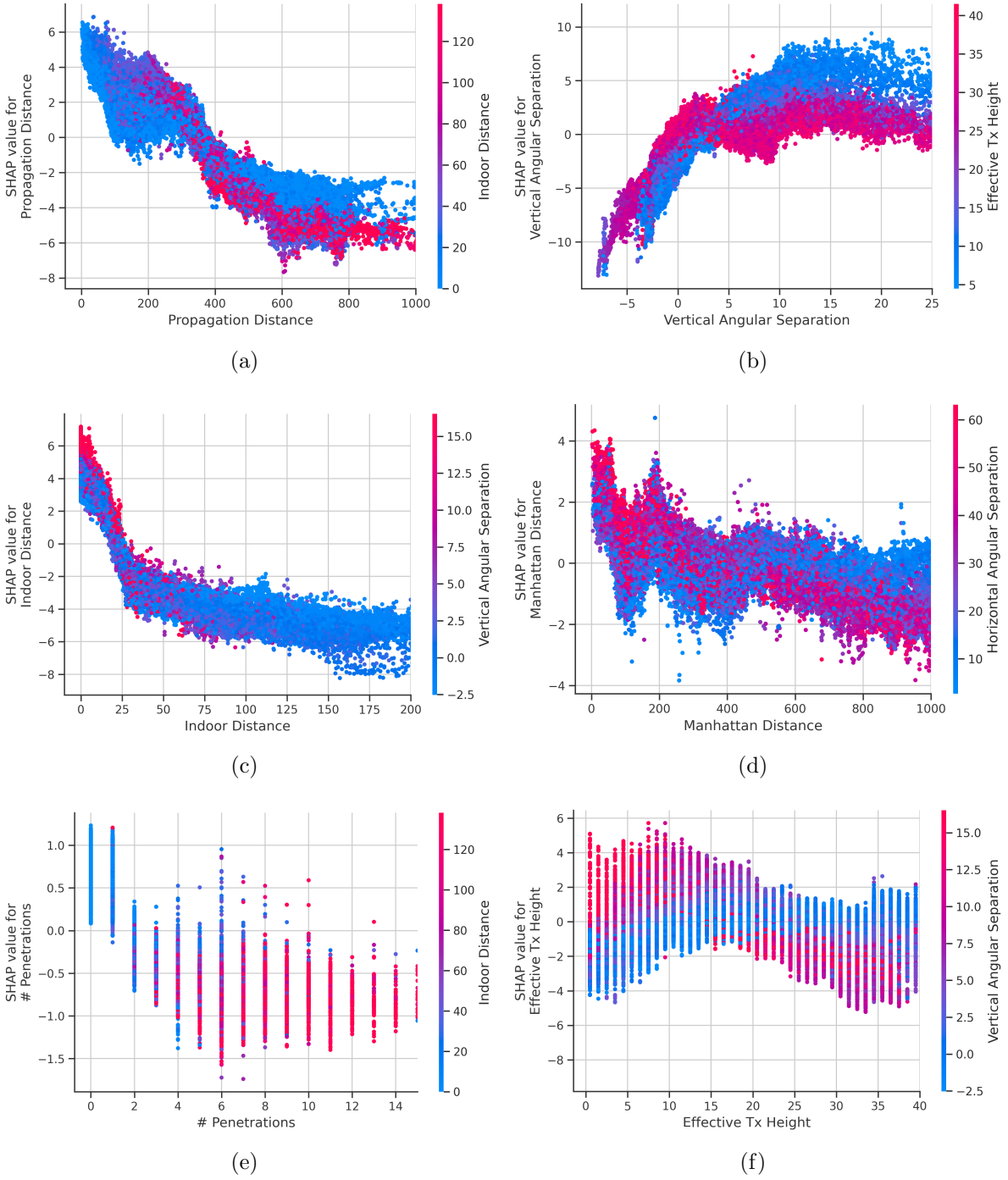


Figure 2.9: Impact of various features and their inter-dependence in predicting Received Signal Strength. Useful for model interpretability and finding important feature regions (or values) for intelligent data collection and network automation

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.

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 Figure 2.9), 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 Figure 2.9(a), 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. Figure 2.9(b) 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 Figure 2.9(c), we can see the increase in feature importance of *indoor distance* at points where vertical angular separation is high. Figure 2.9(d) shows that feature importance of *Manhattan distance* and its interplay with horizontal angular separation. In Figure 2.9(e), 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 (Figure 2.9(e)). Lastly, Figure 2.9(f) shows the feature importance of effective BS height and its interaction with vertical angular separation.

2.4.3 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 Figure 2.8 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 Figure 2.9), 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, Figure 2.9(c) shows that the *indoor distance* and *vertical angular separation* have the highest impact on the model output when $0 < d_{indoor} < 20$ m and $\phi_{ver} > 10°$.

2.4.4 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 Figure 2.9(c), 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 $\phi_{ver} > 10°$ (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 2.6) show that by using the top 5 features (from Figure 2.8(b)), 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 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.

Table 2.6: 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	$+\mathbf{64.07}\%$
Prediction Time	0.077 ± 0.005	0.023 ± 0.002	$+\mathbf{70.13}\%$

2.5 Conclusion

In this chapter, 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.

CHAPTER 3

Moving Towards Few-Shot Learning: A Novel Deep Transfer Learning Approach to Create Network Behavior Prediction Models using Sparse Data

3.1 Introduction

In previous chapter, an AI-based framework is proposed for creating a network behavior model. However, despite their immense potential, the training process of AI/ML-based network behavior models is limited by the sparsity and uneven distribution (unrepresentativeness) of real network data collected by the operators, especially for system-level network modeling. This is because operators only dedicate a few cells in the network for system-level experiments. Experimenting with a large number of base stations to gather data is very costly and can degrade the performance of the network, thus affecting the user experience. Using deep transfer learning, the knowledge learned from a cell can be used to generate data in areas where little or no data is available. TL is a type of machine learning that leverages and synthesizes the distilled knowledge from similar tasks or past experiences to facilitate the learning of new problems [43]. It is inspired by the ability of naturally intelligent species, like humans, to apply knowledge learned from a previous task or domain to some other related tasks. By adopting this, the existing model can be fine-tuned with little or no data, and the model's performance is greatly improved in new or unforeseen situations. TL can address many challenges faced by conventional ML techniques in wireless networks, such as data scarcity and unrepresentative, dynamic variations in the environment, and concerns about security and privacy [44].

3.1.1 Related Work

In the literature, several works have been conducted using transfer learning to address various issues related to network configuration, operation, and performance prediction [45, 46, 47, 48, 49, 50, 51, 52]. To address the problem of parameter configuration for uplink power control and user scheduling using cell KPI/counter data in cellular networks, the authors in [45] use the transferable contextual bandit algorithm, a type of reinforcement learning that is well suited for problems that have many possible actions, and it is not feasible to explore all of those actions. In the context of cellular networks, the authors have formulated the problem of parameter configuration as a sequential decision-making process where each cell needs to select an appropriate set of parameters based on its current context. The authors have developed a collaborative learning-based approach to learn a policy that can be transferred across different cells, resulting in improved decision-making efficiency. Live field tests in a real cellular network consisting of 1700 plus cells show a significant performance improvement of 20% by optimizing five parameters for two weeks, thereby demonstrating the effectiveness of the proposed scheme.

In [46], the authors also tackle the issue of many parameter configurations in cellular networks by adopting the deep neural network-based TL framework to reconstruct radio maps corresponding to a target antenna tilt configuration by transferring knowledge from another tilt configuration of the same antenna. It is assumed that sufficient data is available for the later tilt configuration, relating it to the RSRP values. This data set contains the geographical information of the RSRP measurement position and relevant physical cell identifier information. The source DNN model uses this data set to learn the relationship between the antenna tilt and the RSRP. To determine the best source model, the authors adopt Bayesian optimization to perform the hyperparameter search, i.e., to determine the number of layers, the number of neurons in each layer, the learning rate, and the activation function. The study finds that the performance of predictive models is dependent on the amount of data taken from the testing domain for training

or fine-tuning.

In the study [47], the authors address the challenge of predicting key performance indicators (KPIs) in mobile radio networks with limited data and a high volume of predictions required within a short time frame. They propose a deep transfer learning approach for time series prediction on channel quality indicators and the active number of user equipment (UE) in a cell. The approach involves training a deep neural network on a large data set of multiple cells and frequencies, followed by fine-tuning it on a smaller data set of a target cell or frequency. This enables accurate predictions of KPIs such as CQI and cell load, i.e., active UEs. TL experiments are conducted using various cells in the source domain to train the models. TL-based schemes outperform non-transfer approaches when training data is scarce, leveraging knowledge from multiple cells and frequencies to improve prediction accuracy. Time-series prediction allows for anticipation of future KPI values, enabling proactive measures for network optimization. The study also explores TL optimization and the impact of hyperparameters on model performance, including layer numbers, neurons per layer, and activation functions. It is concluded that exceeding a certain number of layers does not enhance accuracy and may increase computation time. Similarly, an optimal number of neurons per layer is crucial to avoid overfitting. Activation functions like ReLU and ELU outperform sigmoid or tanh. The proposed approach can be extended to other KPIs and offers insights for TL optimization.

The work in [48] aims to minimize energy consumption in cellular radio access networks (RANs) without relying on precise forecasts of dynamic traffic loads. Instead of relying on precise forecasts of traffic loads, which can be difficult to obtain in practice, the authors propose a reinforcement learning framework that models traffic variations as a Markov decision process (MDP). This approach allows the system to adaptively match base station (BS) switching operations with traffic load variations, thereby reducing energy waste and improving network efficiency. This approach has the potential to significantly reduce operational costs and environmental impact while maintaining or even improving

network performance. To speed up the learning process, the paper introduces a transfer actor-critic algorithm (TACT), which utilizes TL expertise from historical periods or neighboring regions. By transferring the learned BS switching operation strategy at historical moments or neighboring regions, TL could take advantage of the temporal and spatial relevancy in traffic loads and speed up the ongoing learning process in regions of interest (target tasks). This approach can improve the efficiency of energy consumption in RANs without relying on precise forecasts of dynamic traffic loads.

To address the challenges faced by 5G networks, such as varying traffic demands and the widespread deployment of different access points, the authors of the study [49] investigate the potential of machine learning techniques and knowledge transfer to optimize network configurations. The focus is on predicting tilt-dependent received signal strength maps, considering two scenarios: transferring knowledge from one tilt configuration of an antenna to another tilt configuration of the same antenna, and transferring knowledge from one tilt configuration to a different antenna with the same tilt setting. The authors compare various transfer learning approaches, including fine-tuning all layers, fine-tuning only the last layer, and freezing certain layers while fine-tuning others. These approaches are compared against models trained separately from scratch for each network configuration. The results demonstrate the efficacy of transfer learning in optimizing network configurations and improving the performance of machine learning models to predict signal strength maps in 5G networks. However, it is worth noting that our proposed work expands on these findings by considering three parameters, including antenna tilt, and investigating their impact on three key performance indicators (KPIs). Additionally, our work employs a transfer learning framework based on residual error-based learning and analyzes the transfer learning strategy through the selection of source and target cells, exploring KPI trends with respect to relevant parameters.

The work presented in [50] aims to enhance the generalization of deep learning models and reduce their computational cost. The authors propose a novel approach to data

augmentation in the embedding space, which replaces the traditional augmentation in the raw input space. This approach offers the benefits of reduced computational costs while maintaining model accuracy. The method involves using pre-trained models on similar datasets through transfer learning. Initially, a large amount of data is used to train the model, which can be done on a cloud platform to alleviate computational concerns. Subsequently, the pre-trained model is fine-tuned to fit the new problem. The results demonstrate that the proposed data augmentation in embedding space significantly reduces the computational cost of training deep learning models while preserving accuracy. Transfer learning and data augmentation techniques prove to be effective in improving model generalization, especially when labeled datasets are limited. Furthermore, it is concluded that fine-tuning a transfer model with data augmentation in the raw input space incurs high computational costs, which can be addressed by utilizing an approximate augmentation method in the embedding space. This proposed approach is particularly valuable for implementing large models on embedded devices with restricted computational and energy resources.

In [52], the authors propose a transfer learning approach to maintain the accuracy of a performance model in a dynamic cloud environment without compromising service performance. They address the challenges of retraining models due to changes in the operational environment during execution. The approach involves retraining a limited number of layers using knowledge from a different environment (source domain), reducing data collection overhead. Real-world cloud service data from a testbed is used to evaluate the performance of the approach for different services under varying loads. The authors compare the accuracy of the prediction with a baseline model trained from scratch using data from the target environment. Results in terms of mean absolute error (MAE) and root mean square error (RMSE) demonstrate that the transfer learning approach significantly reduces data collection overhead while maintaining prediction accuracy.

All the work discussed above has used TL in wireless networks. However, the focus of

this dissertation is network behavior modeling, where the available data in the target environment are extremely sparse and unrepresentative. In these situations, traditional deep-transfer learning does not seem to capture the changes in the target environment from extremely few training data samples.

3.2 AI-based Network Behavior Modeling

Network behavior models based on mathematical and analytical techniques suffer from inaccuracies stemming to oversimplification and impractical assumptions. On the other hand, AI-based models have the potential to assimilate complex and highly nonlinear relationships from data, making them suitable for learning the behavior of dynamic cellular networks. However, developing a comprehensive network behavior model that encompasses the relationships between hundreds of configuration and optimization parameters (COPs) and key performance indicators (KPIs) is an unattainable task, even with the use of AI. In order to reduce the size of the modeling problem to a degree addressable by AI, in this chapter, we focus on the most significant subset of parameters that have the greatest impact on network performance, and that can be used to calculate higher-level KPIs. Specifically, we consider BS height, transmit power, and antenna tilt as COPs, and RSRP, RSRQ, and SINR as KPIs, as shown in Fig. 3.1. Higher-level KPIs such as coverage, throughput, latency and energy efficiency can be determined based on these primary KPIs.

3.2.1 System Model

We consider a cellular network comprising of N base stations, where each base station consists of three sectors/cells deployed to serve the users in their vicinity. Each sector has a dedicated Kathrein directional antenna at a height h from the ground having a downward tilt of ϕ degrees. The transmit power of the antenna is assumed to be P_{tx} and can have a maximum value of 43 dBm depending upon the FCC regulations for that

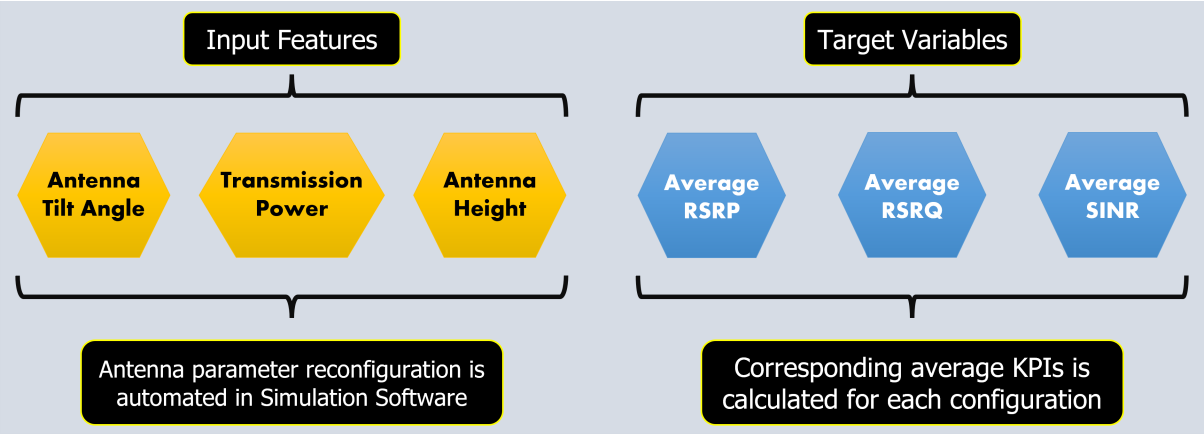


Figure 3.1: Input features and output target parameters for the network behavior prediction modeling.

particular frequency band. The base station antenna can be configured to operate at different values of heights h_n above the ground, ϕ_n different tilt configurations usually ranging from 0 to 10 degree and P_n different transmit powers.

In the geographical region, the base stations serve the UEs which are distributed according to the Poisson distribution. Realistic traffic maps are used to model the traffic demand of UEs in the simulation area. Furthermore, we assume a realistic ray tracing-based propagation model called Aster and clutter-dependent shadowing. The ray-tracing based model leverages digital land use maps with n different clutter types.

Cell-level Configuration Parameters

The key cell-level configuration parameters or data features used to train the network behavior prediction model are defined below:

- **Transmit Power -** represents the maximum transmit power (in dBm) from the base station.
- **Antenna Tilt Angle -** represents the vertical angle from the horizontal plane of the antenna boresight.

- **Antenna Height -** represents the vertical height from the ground elevation to the antenna on the base station.

Making changes to any of the parameters mentioned above can have a contrasting effect on network behavior. For example, increasing the transmit power or antenna tilt expands the coverage area, but it also increases interference levels with neighboring cells.

Cell-level Key Performance Indicators

The key performance indicators for the network behavior prediction modeling are discussed below.

Average RSRP: The average RSRP for a single UE is calculated as,

$$E_{DLRS}^{TX(xc)} = EPRE_{DLRS}^{TX(xc)} + G_{Ant}^{TX} - L^{TX} - L_{Path} - M_{Shad-Model} - L_{Indoor} + G^{M} - L^{M} - L_{Ant}^{M} - L_{Body}^{M}, \quad (3.1)$$

where $EPRE_{DLRS}^{TX(xc)}$ is the energy per resource element for the downlink reference signals, G_{Ant}^{TX} is the transmit antenna gain, L^{TX} is the transmitter loss, L_{Path} is the path loss, $M_{Shad-Model}$ is the amount of shadowing experience by the user at a particular point. L_{Indoor} is the indoor loss, G^{M} is the mobile antenna gain, L^{M} is the mobile loss between the RF chain and mobile antenna, L_{Ant}^{M} is the mobile antenna loss, L_{Body}^{M} is the body loss of mobile holder.

It should be noted that (3.1) and the subsequent equations give the respective KPI values for a single user. However, we are interested in determining the cell level KPIs; hence, we average over all UEs in a particular cell to determine the cell level KPI.

Average RSRQ: The average RSRQ for a single UE is given as

$$RSRQ^{TX(ic)} = E_{DLRS}^{TX(xc)} - RSSI^{TX(xc)} + 10 \times Log(N_{PRB}^{TX(xc)}) + 10 \times \log\left(\frac{W_{Channel}^{TX(xc)}}{W_{Max}^{M}}\right) \quad (3.2)$$

where $E_{DLRS}^{TX(xc)}$ is the RSRP computed using eq. (1), $N_{PRB}^{TX(xc)}$ is the total number of PRBs defined in the frequency bands table for the channel bandwidth used in the cell, $W_{Channel}^{TX(xc)}$ is the channel bandwidth, W_{Max}^{M} is the maximum bandwidth supported by the UE category. $RSSI^{TX(xc)}$ is the received signal strength indicator computed using the following equation,

$$RSSI^{TX(xc)} = 10Log\left(\epsilon_{RSSI}^{TX(xc)} + \sum_{All\ TX(xc)}\left(10^{\frac{\epsilon_{RSSI}^{TX(xc)}}{10}}\right) + I_{DL}^{Inter-Tech} + 12 \times 10^{\frac{n_{Sym}^{TX(xc)}}{10}}\right) + NR_{DL}^{Inter-Tech} + 10 \times Log\left(N_{PRB}^{TX(xc)}\right) + 10 \times \log\left(\frac{W_{Max}^{M}}{W_{Channel}^{TX(xc)}}\right), \quad (3.3)$$

where $I_{DL}^{Inter-Tech}$ is the inter technology downlink interference, $n_{Sym}^{TX(xc)}$ is the downlink noise for one resource element, $NR_{DL}^{Inter-Tech}$ is the inter technology downlink noise rise. Similar to the previous case, the average RSRQ of a cell is determined by averaging these values over all UEs in a particular cell.

Average SINR: The downlink SINR for a single UE in LTE networks is calculated as

$$\gamma = \frac{\overbrace{P_s}^{\text{Signal Power from Serving BS}}}{\underbrace{\left(\sum_{\forall n \in D_{BS} \setminus s} \frac{N_s^{ant}}{N_n^{ant}} P_n r_o\right)}_{\text{Interference Power due to Neighboring BS's}} + \underbrace{\overbrace{1000kTB}^{\text{Thermal Noise}} \times \overbrace{NF}^{\text{Noise Factor}}}_{\text{Noise Power}}}, \quad (3.4)$$

where γ is the SINR experienced by the user associated to BS_s, P_s and P_n are the RSRP (in watts) from serving and neighbor cells respectively, N_s^{ant} and N_n^{ant} are the number of antenna ports in serving and neighbor cells BS transmitter respectively, r_o is the total overlap ratio or the interference reduction factor due to co-channel and adjacent channel

overlap, k is Boltzmann constant, T is the temperature, B is the Bandwidth of sub-carrier, NF is the Noise Factor. The operator '\' in '$D_{BS}\backslash s$' means "elements of D_{BS} except s". Again, averaging the SINR for all UEs in a cell yields the cell level average SINR.

In the following section, we discuss a persistent and crucial challenge encountered in machine learning, i.e., data scarcity and insufficient data for model training. We will also present our proposed framework, which utilizes transfer learning boosted by error residual-based learning, to tackle this issue and enable rapid learning of network behavior models with limited data.

3.3 HybridDeepBoostTL Framework for Extremely Sparse Datasets

Training data-driven AI-based models can be challenging due to data scarcity, particularly when certain cells in a network are newly deployed and lack sufficient data for training their own network behavior models. One effective approach to address this issue is via Transfer learning, where a model trained on a data-enriched site (source site) can be deployed on another site (target site), as shown in Fig. 3.2. The limited data available at the target site can be used to fine-tune the source model. This approach allows for the transfer and generalization of knowledge from one part of the network to other parts, giving operators a network behavior model for the complete network, quickly and efficiently. Therefore, by transferring the knowledge of network behavior from one part to another, operators can quickly determine the optimal network configuration without the need to invest time, effort, and financial resources in conducting drive tests to gather extensive data or acquiring computational resources for training new models.

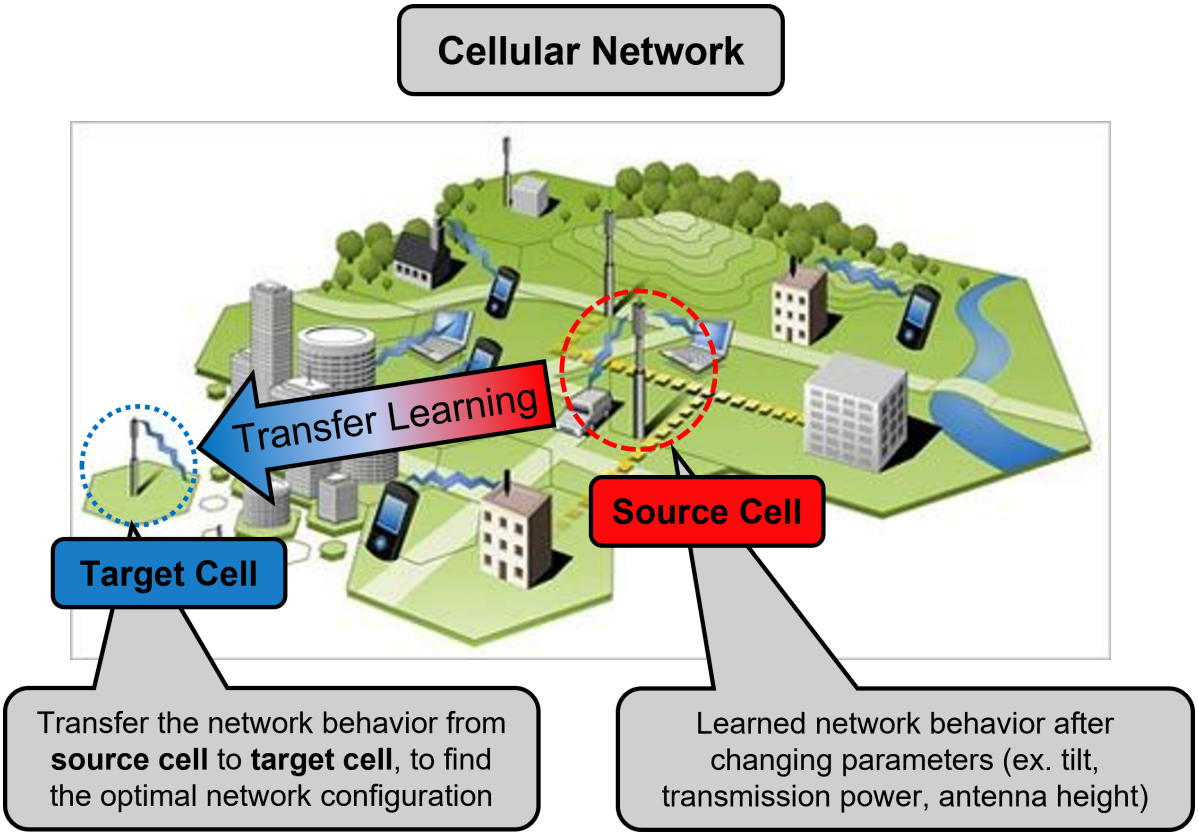


Figure 3.2: An example of using transfer learning in cellular networks for network behavior prediction to find the optimal network configuration in a target cell by transferring the learned network behavior from the source cell (with rich data) to the target cell (with scarce data).

3.3.1 Challenges in Transfer Learning

Transfer learning involves adapting a pre-trained model to a new task or domain by fine-tuning its parameters using new data. To ensure and achieve a high-quality transfer, the following aspects necessitate careful consideration:

What to Transfer: Not all knowledge learned from the source domain would be useful for the target domain. Some knowledge would be common in both the source and target domain, while others may be irrelevant to the target domain. Hence, it is important to identify the relevant features, tasks, and models between the source and target cells. In this dissertation, both the source and target model have the same task and feature set i.e. learn the relationship between the respective cell-level network COPs and KPIs. The former includes the antennas' transmit power, tilt, and height, whereas the latter includes

the average values of RSRP, RSRQ, and SINR.

Where to Transfer: The decision about where to transfer is also important, especially while in mobile cellular networks. The selection of an appropriate target is important because, despite their correlation, not all deployment scenarios in the cellular networks are good for the transfer case. Hence, if the target has a very low correlation with the source, it might not be able to learn anything meaningful from the source model and might result in a negative transfer learning gain.

How to Transfer: Deciding the transfer learning method and its design is another important aspect to achieve a working solution with better performance. Typically, the transfer decision depends on the similarity between the tasks, the pre-trained model's availability and size, and the new task's size and complexity. For example, if the pre-trained model is trained on a task very similar to the new task, it may make sense to use most or all of the pre-trained layers and fine-tune them for the new task. On the other hand, if the pre-trained model is much larger and more complex than the new task, it may be more efficient to use only a subset of the layers and fine-tune them more aggressively. We have analyzed this question by discussing the TL performance with respect to TL strategy, i.e., the number of layers of source cell DNN model retrained on the target cell, in Section 3.4.4. It is shown there that in most cases, the best approach is to retrain merely the last layer of the source cell DNN model as it can achieve significantly good performance.

3.3.2 Conventional TL Framework

The conventional TL framework, which is the starting point for the proposed framework, is shown in Figure 3.3. The training phase is shown on the left whereas the inference phase is shown on the right. During the training phase, based on the abundant data available at the source cell, a DNN model is trained to learn the COP-KPI relation. The COPs (transmit power, antenna tilt, and antenna height) are considered input features while the

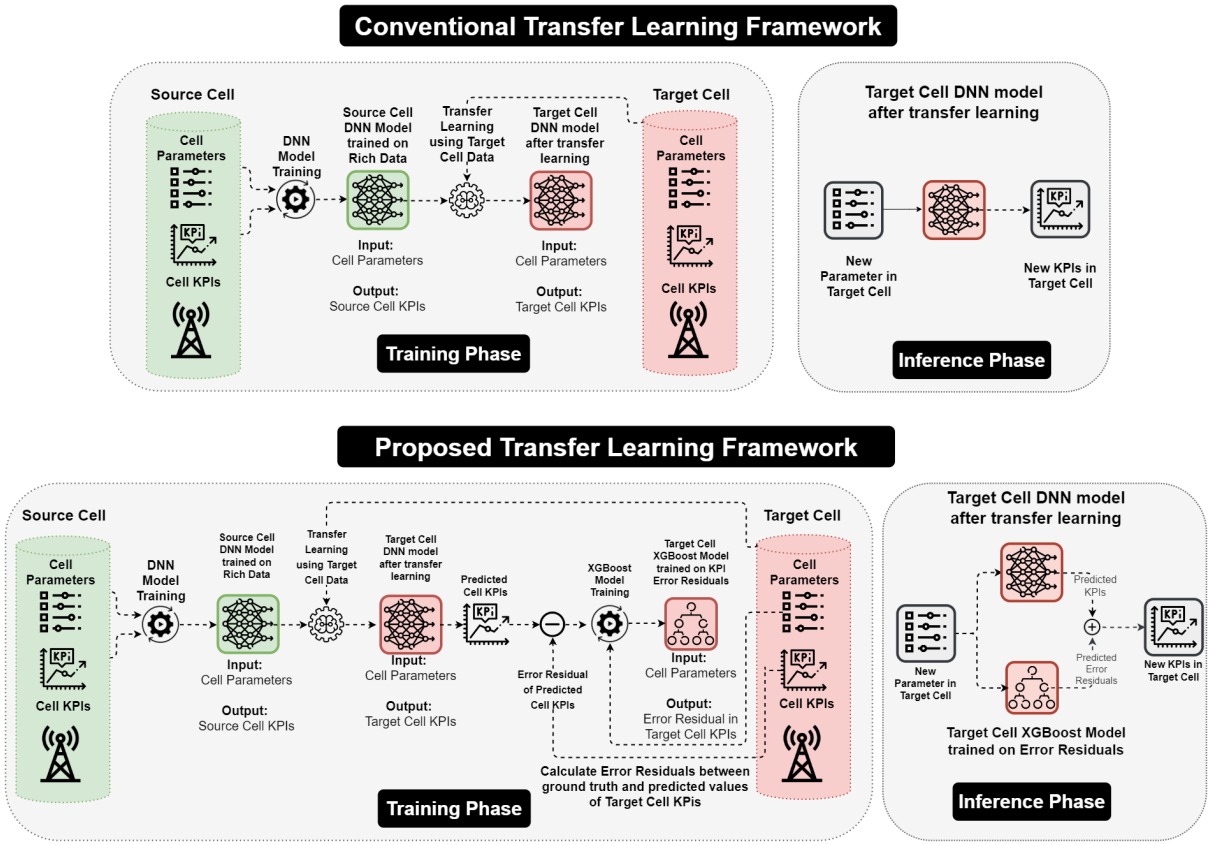


Figure 3.3: The proposed transfer learning framework combines Deep Neural Network and Extreme Gradient Boosting models in a boosting cascade ensemble. This approach is distinct from state-of-the-art deep transfer learning, which simply fine-tunes the source cell model from a data-rich cell. Instead, we calculate the errors of the DNN-based source cell model using a small amount of ground truth data in the target cell and train an ML-based XGBoost model to learn the error behavior. This error model captures where the source cell model fails and augments it in those instances. At inference time, the output from both models is added to give the final prediction, which is helpful in sparse training data scenarios, as the error model does not need to learn the entire network behavior, requiring significantly less training data in the target cell.

KPIs (RSRP, RSRQ, and SINR) serve as labels. Subsequently, the trained source model undergoes fine-tuning using limited data from the target cell. This fine-tuning process involves retraining the last few layers of the source model and updating the weights. The resulting fine-tuned DNN model for the target cell is then employed during the inference phase to predict the target cell's KPIs based on input values of transmit power, antenna tilt, and antenna height.

3.3.3 Limitation of Conventional TL Framework

DNNs are known for their high parameter count, which necessitates a significant amount of data for effective training. Even in the context of transferring DNN models, the fine-tuning phase still requires a considerable amount of data, albeit less than what was needed for the initial training of the source model. Consequently, the transfer learning approach may not deliver optimal results when dealing with *extremely sparse datasets* at the target cell. In such scenarios, the parameters may not receive sufficient updates during the fine-tuning phase, leading to subpar model performance.

3.3.4 Proposed Transfer Learning Framework

In order to enhance the performance of the transfer learning model fine-tuned with a sparse dataset, we propose a *hybrid boosted cascade ensemble transfer learning framework*, illustrated in Figure 3.3. This approach combines the strengths of deep neural networks (DNNs) and eXtreme Gradient Boosting (XGBoost) to improve the accuracy of the network behavior predictions.

Similar to the conventional TL framework, a pre-trained DNN from a source cell is transferred to the target cell and fine-tuned using the extremely sparse target dataset. However, due to the scarcity of data, the fine-tuned model alone may yield inaccurate predictions. To enhance the accuracy of the predictions, we propose augmenting the fine-tuned model with an eXtreme Gradient Boosting (XGBoost) model. This additional model learns and corrects the residual error present in the predictions from the fine-tuned model.

Consequently, during the training phase of our proposed framework, two models are generated: the target cell DNN and the XGBoost model. The XGBoost model is trained to learn the relationship between the COPs as input features and the error in KPI prediction as the model output. In the inference phase, for each input combination of COPs, the

XGBoost model generates a residual signal, which is then added to the KPI predictions of the fine-tuned model to improve the accuracy of the respective predictions.

The rationale behind training an ML model to learn the residual error and correct the output of the fine-tuned model is two-fold: 1) ML models typically have fewer parameters compared to DNNs. Consequently, they can be effectively trained using the extremely sparse data available, thus enhancing the overall performance of the framework. 2) Moreover, this residual learning approach is especially helpful in extremely sparse training data scenarios, as the error model doesn't have to learn the whole network behavior but rather learns the shortcomings of the original source model, thus requiring significantly less training data in the target cell.

The transfer learning strategy (i.e., the number of re-trained layers) and its impact on the overall system performance are discussed in Section 3.4.

3.4 Simulation Setup and Performance Evaluation

3.4.1 Experimental Setup

A ray-tracing-based radio network planning and optimization platform, "Atoll", is utilized to create a large network topology, consisting of 272 base stations and $65,000$ Poisson distributed UEs (User Equipment) in the city of Brussels, Belgium, as illustrated in Fig. 3.4. Atoll simulations are based on the Aster propagation model, which employs advanced ray-tracing techniques to calculate various phenomena that impact radio wave propagation, such as vertical diffraction over roof-tops, horizontal diffraction and reflections based on ray-launching, atmospheric absorption, rain attenuation and vegetation loss, among others. Additionally, the Aster propagation model parameters are pre-calibrated using more than 1.5 million channel measurements from the real environment. For accurate simulations and performance modeling, actual antenna heights and antenna radiation patterns are utilized. The simulation settings utilized in our analysis, which include

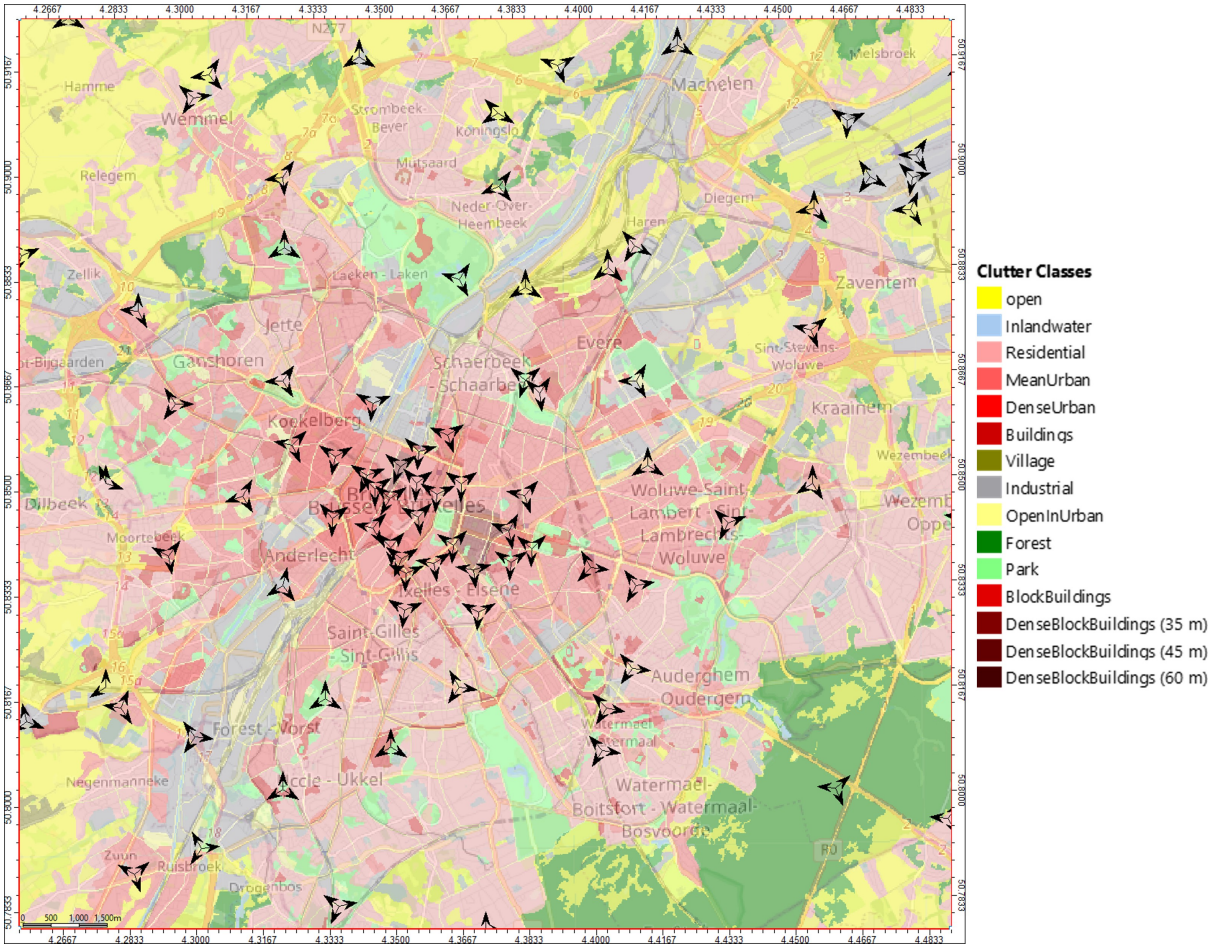


Figure 3.4: Simulation Area in the City of Brussels showing 15 different clutter (land cover) types, e.g., residential, village, forest, park, industrial, open, etc.

system, geographic and eNodeB parameters, are detailed in Table 3.1. Based on the sophisticated propagation modeling and network setting considerations, data calculated by Atoll is taken as ground truth for designing network behavior (COP-KPI) models.

3.4.2 Network Experiments

The range of COP values for the network is given in Table 3.2. The antenna's transmit power was varied from 34 dBm to 43 dBm with a step size of 1 dBm, the antenna down tilt angle changed from 0 to 10 degrees with a step size of 1 degree, and the antenna height was selected from 30m to 45m (above the ground) with a step size of 5 m. This resulted in 400 different COP combinations (10 transmit power values $\times$ 10 tilt values

Table 3.1: Network Scenario Settings

Type	***Parameter***	***Value***
System	Air Technology	4G LTE
	Cellular Layout	81 Macrocell sites
	Sectors	3 sectors per eNB
	Frequency Band	2110 FDD (E-UTRA Band 1)
	Channel Bandwidth	5 MHz
Geographic	Simulation Location	Brussels, Belgium
	Simulation Area	264.76 km^2
	Number of UEs	2068
	UE Distribution	Real Traffic Map-based
	Propagation Model	Aster Propagation Model (Ray Tracing)
	Path Loss Matrix Resolution	10 m
	Geographic Information	Digital Land Use (DLU) Map, with 15 different clutter types
	Shadowing S.D.	$6 - 11$ dB (Clutter Dependent)
eNodeB	eNB Transmit Power (max)	43 dBm
	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^o
	eNB Antenna Vertical Half Power Beamwidth	4.7^o
	eNB Noise Figure	5 dB

$\times$ 4 height values). Consequently, 400 experiments were carried out in both the source and target cells, in order to generate ground truth data for training network behavior models.

Table 3.2: Antenna Parameters reconfigured for the source and target cell and their value ranges during experimentation.

Configuration Parameter	**Value Range**
Tx Power	34 – 43 dBm
Downtilt Angle	0 – 10 deg
Antenna Height	30, 35, 40, 45 m

3.4.3 TL Candidates Selection

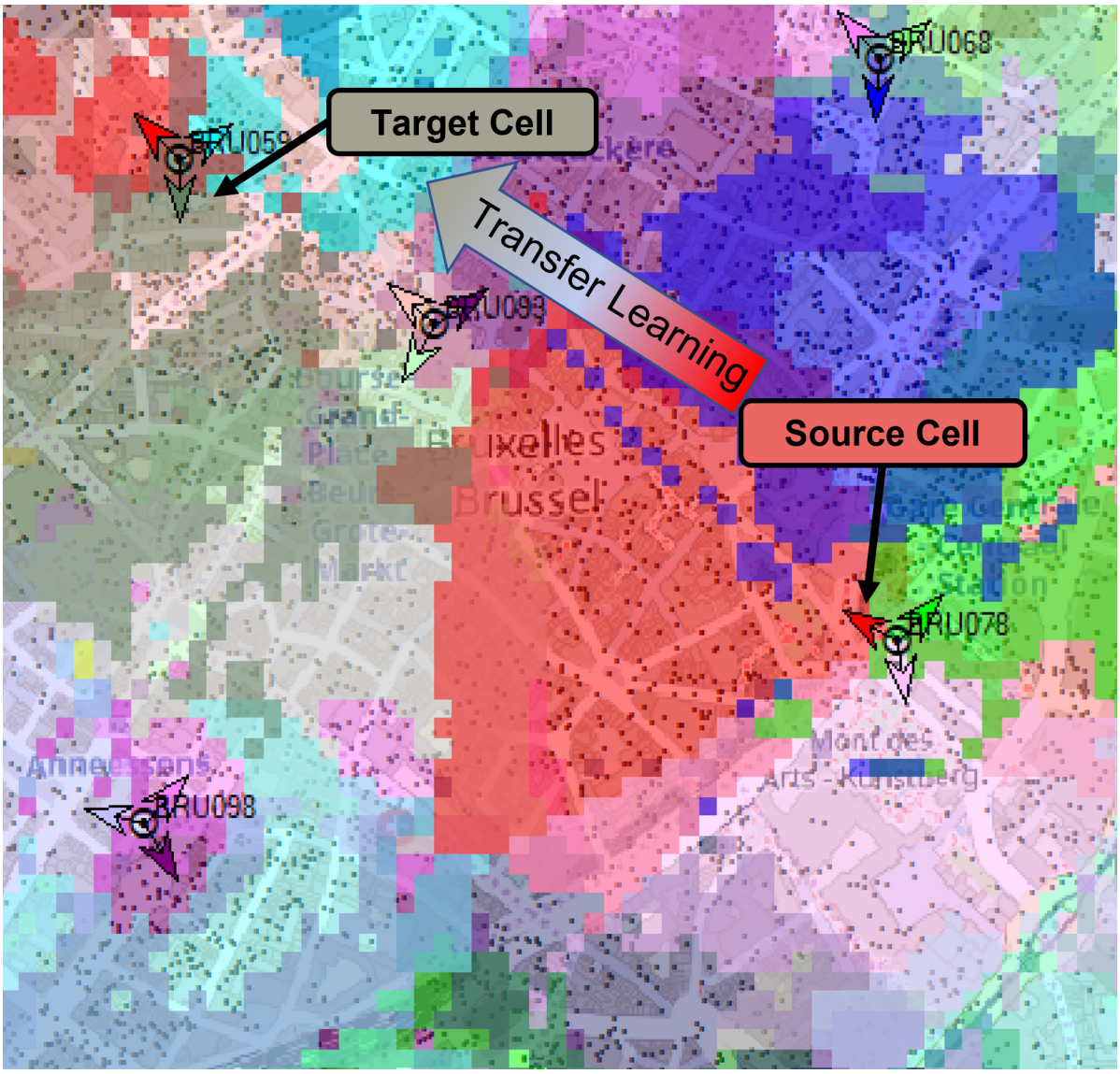


Figure 3.5: A snapshot of the coverage area of the source cell (in red) and target cell (in brown) during simulation experiments. The dots (in black) represent the user distribution.

In this section, we discuss the selection of the source and target candidates for model transfer by examining the distribution of respective KPIs. The source and target cells are shown in Fig. 3.5 and the respective KPI values are presented using contour plots in Fig. 3.6.

Fig. 3.5 shows the snapshot of the source and target cells' coverage area realistically modeled in Atoll as discussed in the previous subsection 3.4.1. Although both of these

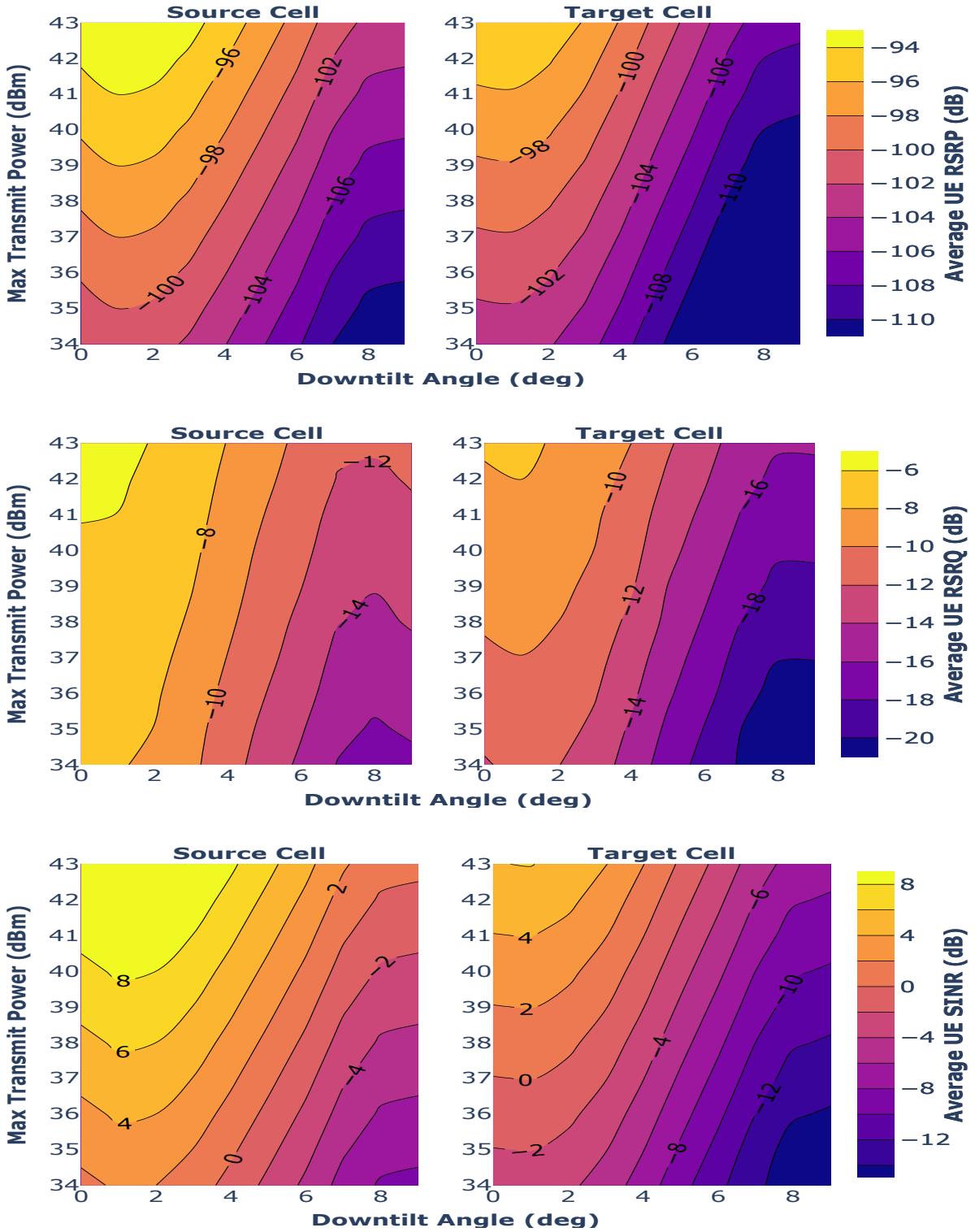


Figure 3.6: Contour plots showing the average RSRP, RSRQ, and SINR of the connected users w.r.t. Antenna Tilt and Transmit Power in the selected source and target cell. The difference between the source cell and target cell KPIs for a given antenna configuration highlights their differences in RF channel conditions, user density, and traffic patterns, therefore making them good candidates for testing the performance of transfer learning from one cell to another.

cells are geographically separated, there exists a significant correlation between their KPIs, as shown in Fig. 3.6 when compared for different network parameters.

In all the subplots of Fig. 3.6, the antenna configuration parameters for both source and target cells comparison is shown by the axes, and the third parameter, showing the respective KPIs, is represented by the color distribution with contours showing the specific values. The correlation between source and target cells for a particular parameter becomes clear by comparing the trend of respective contour curves. Because of the inherently different nature of their data, two respective curves from the source and target cannot be exactly the same; however, their similar shape suggests the presence of high correlation, and hence, the choice of knowledge transfer becomes more viable. Specifically, the contour plots in Fig. 3.6(a) show the comparison of average UE RSRP for a given antenna configuration. For instance, at 43 dBm antenna transmit power and 6 degree down-tilt, the source cell has a -100 dBm Avg. RSRP, whereas the target cell has -104 dBm. Similarly, the contour plots in Fig. 3.6(b) show the difference in average UE RSRP for a given antenna configuration. For instance, the source cell has -11 dB average RSRQ, whereas the target cell has -14 dB for the same antenna configuration parameters. Similarly, the plot in Fig. 3.6(c) shows the difference in average UE SINR for a given antenna configuration. Hence, for the above-mentioned antenna configurations parameter, we get the average SINR of $+2$ dB for the source cell and -4 dB for the target cell. The difference between the source and target cell KPIs for a given antenna configuration highlights their differences in RF channel conditions, user density, and traffic patterns, making them good candidates for testing transfer learning performance from one cell to another. Hence, these two cells are good choices for testing the transfer learning schemes.

3.4.4 DTL strategy and data sparsity impact on TL Performance

In this section, we discuss and compare various strategies of TL by observing the RMSE performance for various network densities. However, before delving into the discussion

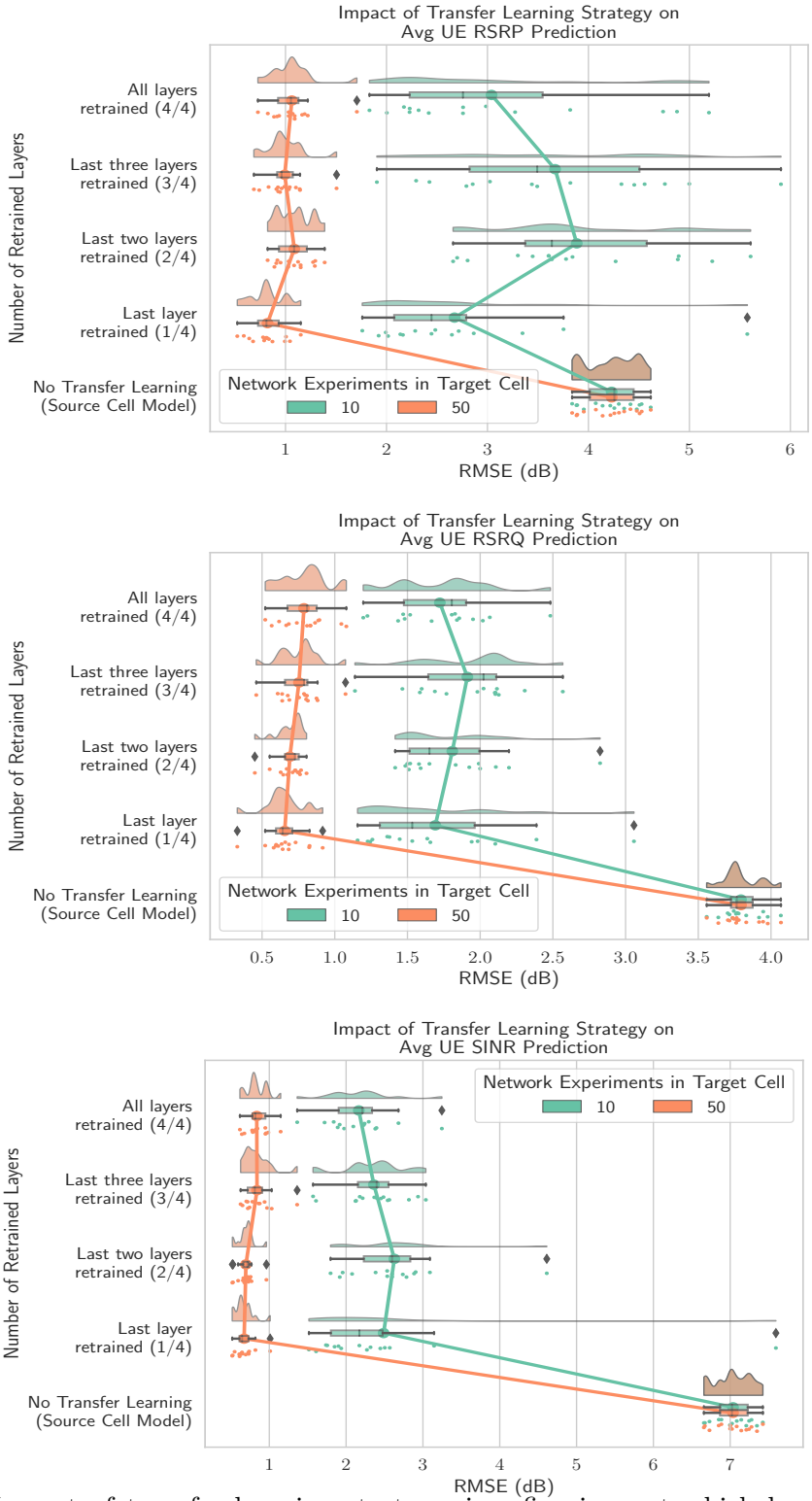


Figure 3.7: Impact of transfer learning strategy, i.e., figuring out which layers of the DNN source cell model should be fine-tuned (re-trained) and which layers should be kept fixed on the prediction performance of the proposed transfer learning framework. Each dot on the RMSE distribution represents a transfer learning experiment from repeated 5-fold cross-validation and hence has a different root mean squared error value from the ground truth.

Table 3.3: Neural Network Parameter Table

Hyper Parameter Name	Hyper Parameter Setting
Layer	4 fully connected layers
Normalization	4 batch normalization layers
Neurons	32 Neurons per layer m
Activation Function	Rectifier Linear Unit
Optimizer	Adam

of results about the TL strategies, we first explain the deep neural network model architecture considered for learning the source and target cells and performing the transfer learning experiments.

We train a deep neural network from the data-rich source cell to model the behavior of average RSRP across all its serving UEs with respect to the transmit power, antenna height, and antenna tilt angle. The relevant parameters of the deep neural network are listed in Table 3.3. Four Fully-connected layers and 4 batch normalization layers are used in the model. This model can be reused in a similar target cell with sparse available data. This is where transfer learning jumps in and fine-tunes the source model using the sparse available data in the target cell. Layer weight transfer and fine-tuning are only done in the fully-connected layers, while the batch normalization layers are always retrained on the new training data. Results in figures 3.7 and 3.8 show that using transfer learning, the prediction RMSE gets reduced by 80% as compared to only the source model without using transfer learning in the target cell.

The transfer learning performance is greatly impacted by the number of layers employed for the retraining in the TL process, which is also described as the DTL strategy in [52]. Specifically, the transfer learning strategy is about figuring out which layer(s) of the DNN source cell model should be fine-tuned (re-trained) and which of those should be fixed to achieve the minimum error in transfer. Also the performance is impacted by the sparsity of data, which is manifested here by the number of experiments considered for

the transfers. Hence, in this section, we discuss the TL model evaluation performance computed as the average value and distributions of training RMSE values with respect to DTL strategy and target data sparsity. Therefore, the performance of model transfer with respect to these two aspects for the respective KPIs is depicted in the subfigures of Fig. 3.7. Unlike the conventional results, in all these subfigures, the dependent variable, RMSE, is shown on the x-axis for a better presentation of error distributions.

Further, in all these subfigures, two cases of target data sparsity are considered, i.e., the *high density* case is about 50 network experiments in the target cell, here the term 'density' is used as opposite to sparsity, whereas *low density* case is about the 10 experiments.

For any case of target data sparsity, knowledge transfer is carried out by retraining a certain number of final layers of the source network. This number varies from 1 to 4, as shown on the y-axis in the subfigures, showing the respective number of last layers trained for the transfer learning. Further, the case of no transfer is also compared as a baseline to show the increase or decrease in prediction RMSE as compared to the basline source cell model without transfer learning. In all the plots, for each transfer strategy, the training error after 15 TL iterations of repeated 5-fold cross validation are shown. The distribution of RMSE is also plotted along with a boxplot to observe the generalized error distribution/trend. The solid line connecting the center of box plots represents the Mean RMSE.

In all these results, the first important observation to note is the comparison of RMSE performance of TL strategies with that of *no transfer learning* as a baseline, where the source model is directly used for testing on the target data, without any retraining. It is clear from the figures that every strategy of transfer learning for every target data sparsity case yields better performance than the source cell model, hence the situation of negative transfer, where the transfer learning, in fact, results in the worst performance, has not occurred here. Hence, it is safe and much better to adopt transfer learning regardless of the target data sparsity or the considered TL strategy than not adopting it at all.

However, the TL gain is significantly lower for the UE RSRP than the RSRQ and SINR cases.

The second significant point to observe is that for all KPIs under consideration and for all TL strategies, the RMSE performance of the *high density* cases is much higher than that of *low density* counterparts. This observation is also in line with the fundamental working of DNN models where training on the highly sparse data results in the model under-fitting which eventually results in higher RMSE values in the testing phase. This aspect of conventional DNN model training is observed in the transfer learning as well, because the retraining of some layers of the source model on target data is essentially similar to the conventional DNN model training, at least from the perspective of model fitting. Similarly, the distribution of the RMSE values in the sparse cases is also higher. Hence, from this point, it can be concluded that although the TL gain is considerable even with the low density, its significant, and consistent RMSE improvement is observed with a relatively high density of target data.

Another important aspect to analyze in these results is about identifying and discussing the best TL strategy, i.e., how many layers should be trained to achieve the best RMSE performance. It is observed from all these figures, that mostly the best RMSE performance is achieved when only one layer of the source model is retrained, in both cases of target data sparsity. This trend is consistently followed in the high-density case, whereas in the low-density case, we observe a slight deviation for the case of UE SINR which will be discussed in a while. However, this general phenomenon of achieving the best RMSE performance with the minimum possible layer of retraining requirements further bolsters the adoption of transfer learning as this requires the minimum training time and computational complexity.

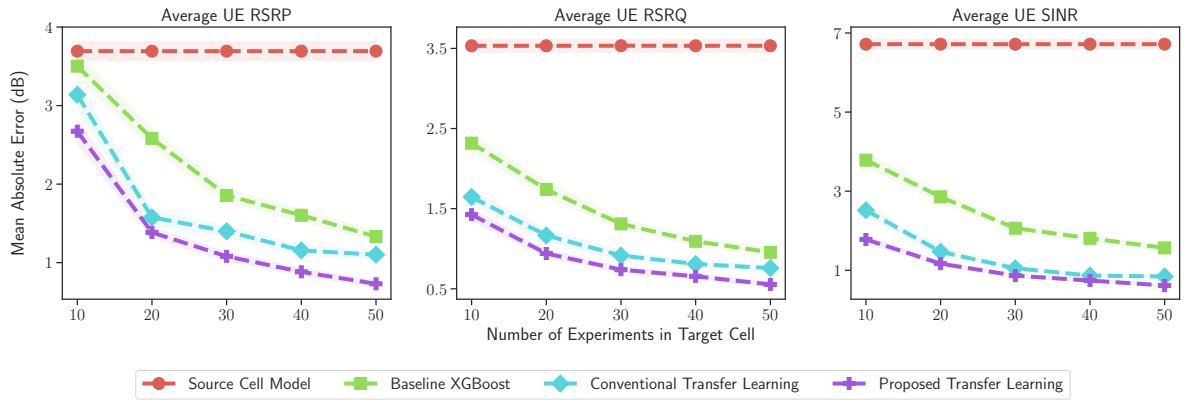


Figure 3.8: Performance comparison of the proposed transfer learning framework with *source cell model* (no transfer learning from data-rich source cell), *XGBoost model* (trained on sparse target cell data), and *conventional transfer learning* (source-cell DNN model fine-tuned on target cell data. Extensive transfer learning experiments show the superior performance of the proposed approach against baseline approaches for predicting the mean RSRP, mean RSRQ, and mean SINR of the UEs in the footprint of the target cell, across various data sparsity levels (i.e, the available number of network experiments in target cell). The dotted line represents the mean value and the filled area/polygon around it represents the 95% confidence interval of the reported mean absolute error after 5-fold repeated cross-validation.

3.4.5 Comparison with the Existing Models

In this subsection, the comparison of the error performance of the proposed scheme with three different schemes is shown for all KPIs. The comparing scheme is the Source Cell Model representing the no transfer situation, Baseline XGboost, Conventional Transfer Learning using the DNN only, and the proposed TL scheme using the DNN and residual error-based boosting model. The conventional and proposed schemes are trained by training the only last layer of the model, as was shown in Fig. 3.7 and discussed earlier in Section 3.4.4. Each sub-figure is dedicated to the respective KPI, as their titles show. These comparisons are made with respect to the increased density of the number of experiments in the target cell, as shown on the x-axis of the figures.

These results show that the MAS of all TL schemes decreases with respect to the increasing number of experiments, which highlights the better performance of TL with increased target data density. The non-varying performance of the non-transfer case is also observed in all the comparisons, and this case is also having the worst performance

of all comparing schemes, justifying the use of transfer learning and proving the positive gains achieved by the transfers. Comparing the TL schemes shows that the XGBoost performs worst in all cases than the conventional DNN and proposed scheme because the TL inherently works by training some layers of deep learning models unavailable in XGBoost. The improved performance of the proposed scheme is also observed in all KPIs cases, when compared with the conventional DNN-based scheme, because of the positive gain achieved by the residual error based boosting.

3.5 Conclusion

In this chapter, a novel method for transfer learning (as compared to state-of-the-art DNN based) is proposed using a boosting cascade-ensemble of Deep Neural Networks and Extreme Gradient Boosting models. The invention is estimated to result in a 25% increase in models' prediction accuracy as compared to baseline transfer learning (DNN-based) if similar amount of training data (network experiments) is used for modeling. On the other hand, this also means less amount of data (network experiments) requirement for giving the same performance, which can result in potential savings for the mobile network operators. Furthermore, we have also performed extensive transfer learning experiments and analyze different transfer learning settings across various data sparsity conditions to draw insights into designing an optimal transfer learning model based on the available data in target domain, which will be further helpful for designing such transfer learning systems in the absence of ground truth data.

CHAPTER 4

Where to Transfer: AI-based Domain-Aware Similarity Metric to Enable Transfer Learning in Emerging Cellular Networks

4.1 Introduction

In the previous chapter, a novel transfer learning framework was proposed to handle sparse and unrepresentative network data. However, real wireless networks are constantly changing in terms of their spatial and temporal properties. This means that the network topology, connectivity, interference, and channel quality vary depending on the location and time of the nodes. These variations pose significant challenges for designing and managing wireless networks, as they affect data transmission's performance, reliability, and security. Also, this dynamicity causes a shift in the independent variable (covariate shift), a shift in the target/dependent variable (prior probability shift), or a shift in the relationship between independent and dependent variables (concept drift) [53]. Due to these shifts in network data distribution with time, the model accuracy reduces over time. In these situations, transfer learning can help, as the existing model can be fine-tuned with little or no data, and the model's performance is greatly improved in new or unforeseen situations. But the selection of similar (source cells) is challenging in a dynamic environment. The traditional similarity (distance) metrics, such as KL, Jensen Shannon, or Wasserstein's metrics, calculate the distance between the estimated data distribution of a (data-rich) source and (sparse) target cell (domain). However, in sparse and unrepresentative data scenarios, we can't accurately estimate the data distribution; hence, the distance or similarity between different cells will also be inaccurate.

4.1.1 Existing solutions and their limitations

Various methodologies have been employed to identify appropriate source and target selections for numerous application domains in Transfer Learning (TL) [47, 54, 55, 56, 57].

In one instance, Parera et al. [47] introduced a deep TL technique to predict Key Performance Indicators (KPIs), specifically the Channel Quality Indicator (CQI) and the number of active User Equipments (UEs) within mobile radio networks. This methodology comprises a two-stage knowledge transfer strategy utilizing information obtained from multiple frequencies and cells. Initially, feature extraction occurs from the data of a particular cell, followed by a fine-tuning stage applied to the same cell's data to enhance prediction accuracy. Despite improving prediction accuracy, this paper does not sufficiently address the question of "where to transfer," meaning the selection of the most suitable source domain(s) based on their similarity to the target domain.

Larsson et al. [58] addressed this question by proposing two strategies for source selection in TL: diversity of the source domain and the similarity between the source and target domains. The authors argue that diverse source domain selection can yield improved performance, especially when the target domain has limited sample availability. Selecting diverse source domains allows for the learning of more generalizable features applicable across various scenarios. Nevertheless, further research is necessary to establish general guidelines for source-domain selection in data-driven modeling supporting networked service management and automation.

In another paper by Pino et al. [54], similarity metrics were utilized to identify analogous financial Markov Decision Processes (MDPs) with existing learned policies, from which knowledge can be transferred to learn new MDP policies. The proposed solution incorporates similarity metrics based on conceptual, structural, and performance relationships among different financial markets. The evaluation demonstrated that employing TL with similarity metrics enhances the efficiency of learning optimal policies for financial MDPs.

Zhang et al. [55] presented a model-based TL strategy that capitalizes on similarities among cellular traffic. Using the high similarity among datasets of SMS, call, and internet traffic, and high correlation coefficients among different types of cellular traffic, the authors employed parameter initialization to transfer knowledge. Specifically, the target domain model's parameters were initialized using those learned from the source domain model, thereby leveraging similarities between different types of cellular traffic and reducing the required amount of training data for the target domain.

4.2 Where to Transfer? A case study to highlight the impact of source cell selection on TL performance

This section presents a case study to show the impact of the selection of source cells on the TL performance, highlighting the importance of a quantifiable strategy for the source cell section (i.e., where to transfer?). For our analysis here, we use the same transfer learning framework, system model and experimental setup as explained in the previous chapter for network behavior prediction modeling.

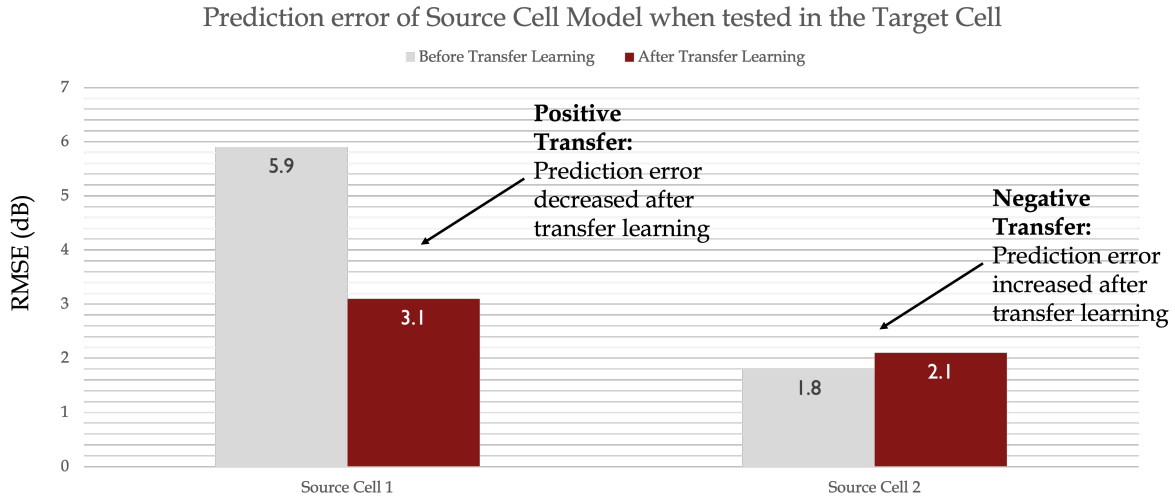


Figure 4.1: Examples of *positive transfer* where the prediction RMSE of the source model is reduced after transfer learning, as well as *negative transfer* where the prediction RMSE of the source model is increased after transfer learning. These examples highlight the importance of source cell selection on the performance of transfer learning.

In the conducted experiments (shown in Fig. 4.1), data from two distinct source cells was utilized for transfer learning in the target cell. Given that comprehensive ground truth data for the target cell was available during our experimentation, it was observed that transfer learning from one source cell resulted in positive transfer. However, when a non-representative amount of COP-KPI experiment data (10) was used for transfer learning from another source cell, it led to a negative transfer. In such an instance, the model, post-transfer learning, exhibited poorer performance than it would have, had we simply used the source cell model for network behavior prediction in the target cell.

This outcome raises a pivotal question: how can we ensure that the transfer learning experiment will improve predictive accuracy when the target cell lacks a representative amount of training data? Existing similarity metrics, such as Jensen-Shannon, Wasserstein, and Energy-based distance metrics, assigned very similar similarity scores to both pairs of source-target cells, making it impossible to predict a priori which source cell would be beneficial for our target cell. This situation underscores the need for the creation of a domain-aware similarity metric that is robust to the sparsity of samples in the target cell.

4.3 ML-based Domain Aware Similarity Metric

Motivated by the importance of the source cell selection criteria for transfer learning, in this section, we discuss the selection based on the domain-aware similarity metric. There can be various criteria for the selection of source cells for TL such as using domain knowledge to identify cells with similar environemnt and network conditions. Another approach can be to apply any dimensionality reduction technique to the cell data and cluster the cells into groups of similar cells. Then for the transfer to a target cell of a certain group, a cell from the same group can be selected as the source cell. Another approach can be to use the distance or divergence metric to quantify the similarity between different cells.

The similarity in machine learning problems is calculated by computing the distance

between data points in a feature space. There are a plethora of distance metrics available in the literature and the choice of the appropriate one is dependent on the type of data and the nature of the underlying problem. For instance, Euclidean, Mahalanobis, and Chebyshev distances are used for the continuous data; and Hamming distance, Jaccard distance, etc., are used for the categorical data. Similarity metrics are also employed to determine the candidate. In order to determine the candidates for TL, similarity metrics are employed in the literature for various domains. However, the conventional similarity metrics are not useful for the problems of network behavior predictions in mobile cellular networks, because of various limitations discussed in the following subsection.

4.3.1 Limitation in Current Similarity Metrics

The TL's selection of source and target cells is decided based on the values of certain similarity metrics between these two. The conventional similarity metrics, such as the Wasserstein and Jensen Shannon, use the data set from both the source and target cells to determine the corresponding metrics' respective values.

Regardless of the underlying working of the adopted similarity metrics, all these schemes work well when the available data for comparison is well representative of the source and target cells. Also, it is assumed that a significant amount of data is available from both the source and target cells under consideration for the transfer. However, these assumptions are hardly realizable in the wireless domain, especially for those not yet deployed or commissioned cells. Hence, these conventional similarity metrics are not feasible for comparing commissioned and operating cells and the proposed new cells because they are not yet generating the same type of data available from the source.

4.3.2 Proposed ML-based Domain Aware Similarity Prediction Framework

The previous section highlighted traditional similarity metrics' limitations when applied to mobile cellular network transfer learning (TL) problems. Consequently, there is a

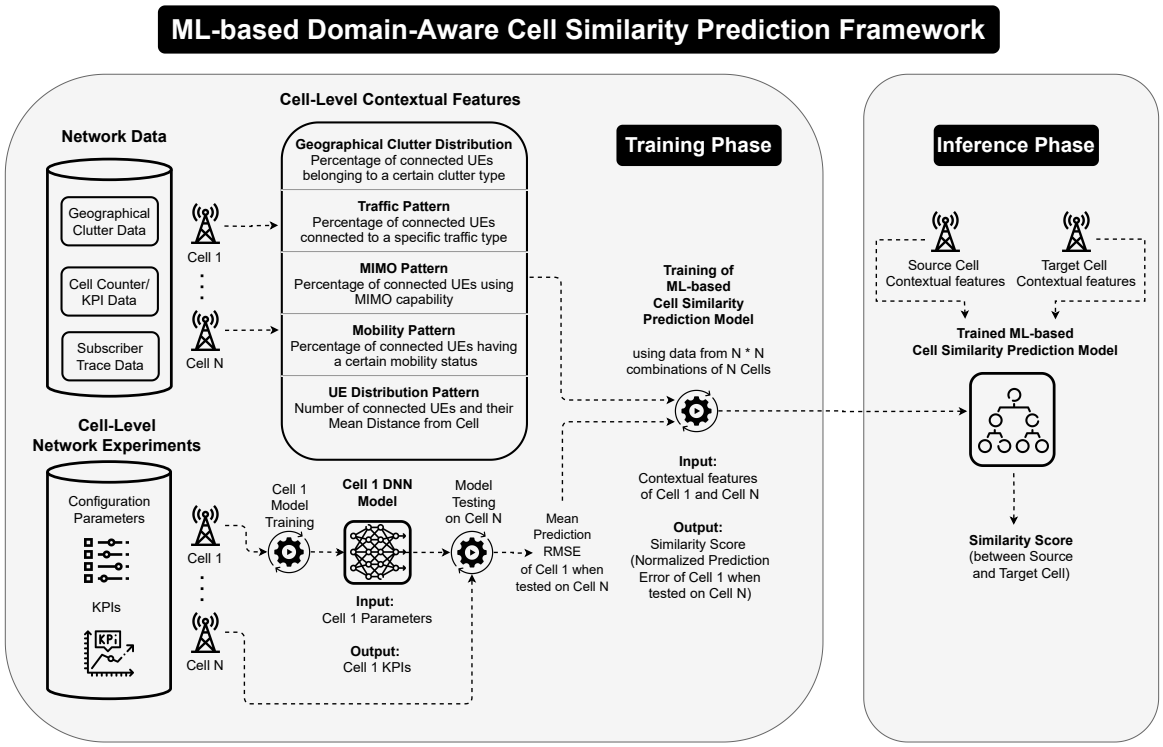


Figure 4.2: A novel AI-based domain-aware cell similarity prediction framework is proposed that is robust to the spatio-temporal dynamics in the network. By leveraging latent features such as geographical clutter pattern which can act as a proxy for spatial dynamics, and cell load, user mobility and traffic pattern etc., which can act as a proxy for temporal dynamics in the network, the proposed framework can measure similarity between cells deployed across different geographical environments and with different user traffic and mobility patterns. The proposed novel set of latent features can be computed for each cell using already collected network data (e.g., cell trace data, geographical clutter data, cell performance management KPI/counter data, etc.). This data is readily available to the operators without the need for further standardization.

growing demand for a domain-aware metric that takes into account the TL benefits obtained during transfers between cells. This section aims to explore how domain-specific parameters can be utilized to devise similarity metrics that incorporate the key aspects of the underlying TL domain. In order to achieve this, we present the framework shown in Fig 4.2, which explains the training and inference phases of the domain-aware similarity prediction. Along with the usual training of the ML model, the training phase also explains cell-level contextual feature extraction from network data of various cells and their categorization, which are used as the input to ML model training. The ML model is trained to learn the TL gain manifested by the Mean prediction RMSE of the respective

cells on which TL is applied.

Mobile cellular network operators possess diverse data sources, such as data about geographical clutter, cell counter/KPI data, subscriber traces, configuration, and performance management data. For most of the network KPIs' behavior modeling and optimization, we utilize the cell-level configuration parameters data, and DNN models are trained to learn the KPIs pattern with respect to these parameters.

Also, most of the transfer learning adoption in the cellular network revolves around utilizing these DNN models trained on various cells as the TL source. Along with the configuration parameters data, each cell in the network encompasses the auxiliary data, which can provide cell-level contextual features which can be exploited for domain-aware similarity metric formulation for transfer learning. The auxiliary data parameters considered for this paper are shown in the framework diagram and enlisted in detail in Table 4.1. They are grouped into five different categories, mainly discussing (1) the percentage of an eNB's coverage area belonging to a certain geographical clutter, (2) the percentage of the UEs exhibiting a traffic pattern, (3) UEs MIMO adoption percentage, (4) ration of the mobile users, and proportions of the UEs having a certain mobility pattern, (5), the pattern of the UEs distribution in a cell. The percentages in the sub-columns of the last column show the proportion of certain features within each subgroup for a given eNB. Hence, the sum of percentages of all features in a group should be 100%. For instance, the traffic pattern of eNB 1 shows that 5% of that is VoIP, 38% is FTP Download, 38% is Video Conferencing, and 19% is Web browsing.

In order to prepare the data for training the domain-aware cell similarity prediction modeling, transfer experiments are carried out between different pairs of cells as depicted at the bottom half of the training phase in Fig. 4.2. In these experiments, the DNN model trained on one cell data, e.g., Cell 1, serves as the source of transfer. This trained model is then gradually tested on the data from other cells, and the corresponding mean prediction RMSE of each respective transfer experiment is recorded against the respective cell-level

contextual features of both cells participating in the experiment. For our experiments, 22 features from each cell contextual data are recorded; hence for each pair of cells in a transfer, the feature vector has 44 values. This process repeats iteratively; hence the DNN model of each cell in the network is employed as the source of the transfer, and its transfer experiment to all the cells in the network is carried out. Overall, we can conduct N by N transfer experiment for a system of N cells, resulting in the N by N instances database, with each instance having 22 features and 1 target.

The cell similarity prediction model is created by training the XGBoost model using the proposed 44-dimensional features as input (22 features each of source and target cell as shown in Table 2). The values for the prediction model's target variable (output) are obtained from testing the source cell model in a given target cell. This RMSE is scaled between 0 and 1, where 1 reflects the minimum prediction error between a source and target cell, representing almost or completely similar cells, and 0 represents the maximum prediction error, thus representing completely dissimilar cells. *It is worth mentioning here that the input features of this proposed cell similarity prediction model are obtained from the auxiliary data that are readily available to the operators for all cells in the network, whereas the respective output data of the model is obtained using the network experiment (COP-KPI) data that is not readily available for all the cells in the network and its availability is very sparse and unrepresentative. Therefore, operators can train this cell similarity prediction model using data from only those cells where the network experiment data is available and correlate it with the readily available proposed auxiliary data (based on geographical, cell counter, and cell trace data).*

For the considered 32 cells system, overall, 1024 possible TL experiments are done, and each experiment result corresponds to the target variable of the one data instance of the resulting database. A trained XGBoost model based on these input-output data will then be able to predict the similarity and, thus, potential transfer learning gain by using the auxiliary features of a new set of source and target cells whose data was unavailable and

not included in the training process. It should be noted that the purpose of the proposed domain-aware similarity metric framework is to give a generic methodology for using the different auxiliary data of different cells, the actual transfers carried on various COP-KPI data, and the resulting performance of these transfers. Hence, it should be taken as the guiding approach, which has shown promising results for the abovementioned auxiliary data parameters and COP-KPI parameters on which transfer learning is done. However, this framework will be applicable to the different parameters of auxiliary data and the COP-KPI data on which TL experiments are done.

Table 4.1: Description of 22 features for each cell in the network. An example of data types collected for each of the 33 cells analyzed in this study

Features			**Cell ID**					
Type	**ID**	**Name**	**1**	**2**	**3**	**.**	**.**	**33**
Geographical Clutter Distribution Pattern (Percentage of an eNB's coverage area belonging to a certain clutter type)	**1**	Mean Urban	19%	69%	18%	.	.	0%
	2	Open in Urban	14%	4%	0%	.	.	0%
	3	Block Buildings	10%	0%	18%	.	.	0%
	4	Park	14%	0%	0%	.	.	0%
	5	Dense Urban	0%	0%	0%	.	.	0%
	6	Residential	0%	27%	9%	.	.	0%
	7	Industrial	43%	0%	9%	.	.	0%
	8	Open	0%	0%	36%	.	.	53%
	9	Dense Block Buildings	0%	0%	0%	.	.	0%
	10	Forest	0%	0%	0%	.	.	47%
	11	Buildings	0%	0%	9%	.	.	0%
Traffic Pattern (Percentage of UEs connected to a specific traffic type)	**12**	VoIP	5%	0%	18%	.	.	13%
	13	FTP Download	38%	42%	55%	.	.	47%
	14	Video Conferencing	38%	50%	27%	.	.	33%
	15	Web Browsing	19%	8%	0%	.	.	7%
MIMO Pattern (Percentage of connected UEs using MIMO capability)	**16**	MIMO Terminal	43%	35%	9%	.	.	27%
	17	Mobile Terminal	57%	65%	91%	.	.	73%
Mobility Pattern (Percentage of connected UEs having a certain mobility status)	**18**	90 km/h	0%	12%	9%	.	.	13%
	19	50 km/h	43%	23%	36%	.	.	40%
	20	Pedestrian	57%	65%	55%	.	.	47%
UE Distribution Pattern	**21**	# Connected Users	21	26	11	.	.	15
	22	Mean Distance between cell and its connected UEs (m)	1109	1199	775	.	.	2573

4.3.3 Performance Comparison with the Existing Models

The terms distance and divergence measures are loosely used interchangeably; however, there is a clear distinction between these two based on the underlying properties. The distance measure plays an important role in ML problems, as it is generally the objec-

tive score to summarize the relative difference between two objects in a problem. For instance, for a typical ML-based regression problem, one object can be the row of the data describing the original value of the RSRP, and the other object can be a similar row describing the predicted values from a trained model. Whereas the divergence measure specifically focuses on comparing the probability distributions. Some of the important divergence measures are discussed in the following subsections.

- To quantify similarities among the cells, one approach is to use the Wasserstein distance measure [59]. Given two random variables f_i and f_j with marginal distributions $P(f_i)$ and $P(f_j)$ respectively, let ψ denote the set of all possible joint distributions that have marginals of $P(f_i)$ and $P(f_j)$. Then Wasserstein distance between them is defined as:

$$W(i,\ j) = inf f_{ij} \in \psi \int |i-j| f_{ij}(i,j) didj. \tag{4.1}$$

- Jensen-Shannon divergence is also a method of measuring the similarity between two probability distributions. Although it is similar to the Kullback-Leibler divergence, it is symmetric and always has a finite value.

$$D_{JS}(f_i,\ f_j) = \frac{1}{2} \int (f_i - f_j)\,(lnf_i - lnf_j)\,dx \tag{4.2}$$

- The energy-based distance measure is defined as the square root of the integral of the product of the difference between the cumulative distribution functions of the two probability distributions raised to the power of two.

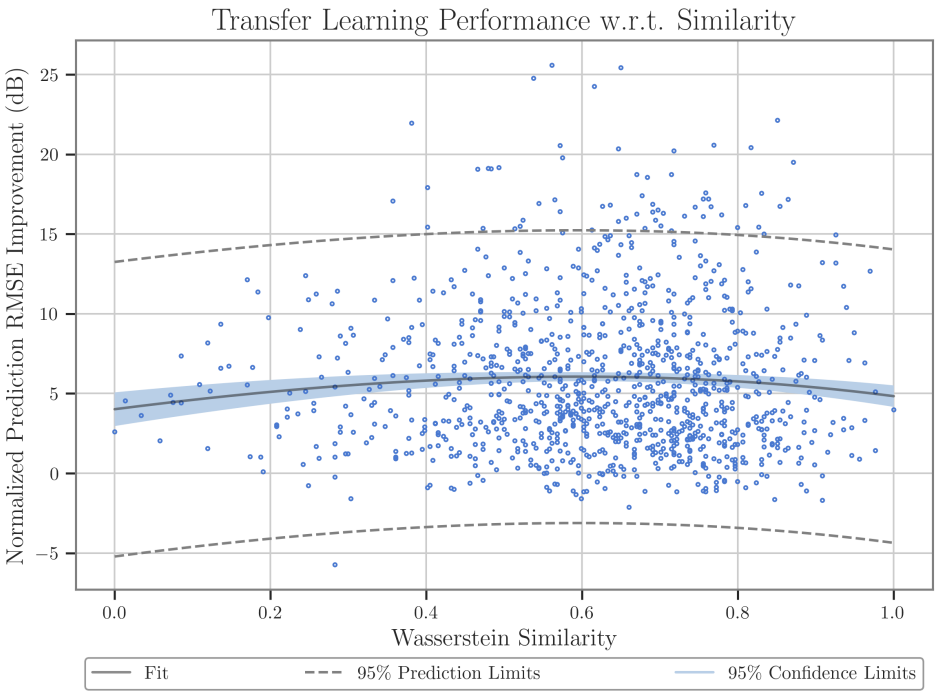


Figure 4.3: Source Cell Model RMSE improvement after Transfer Learning w.r.t Wasserstein similarity.

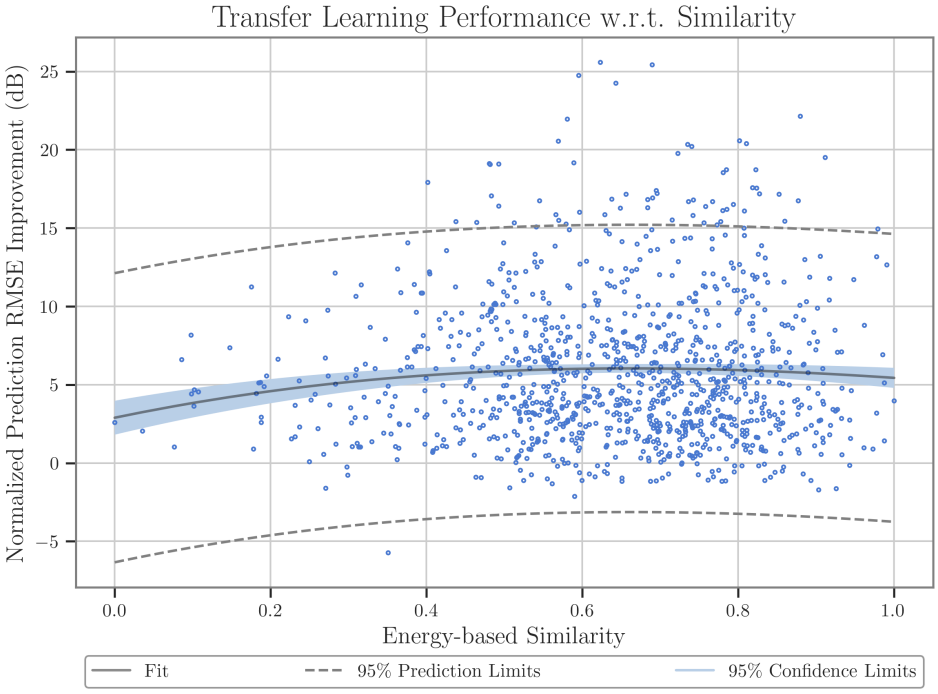


Figure 4.4: Source Cell Model RMSE improvement after Transfer Learning w.r.t Energy based similarity.

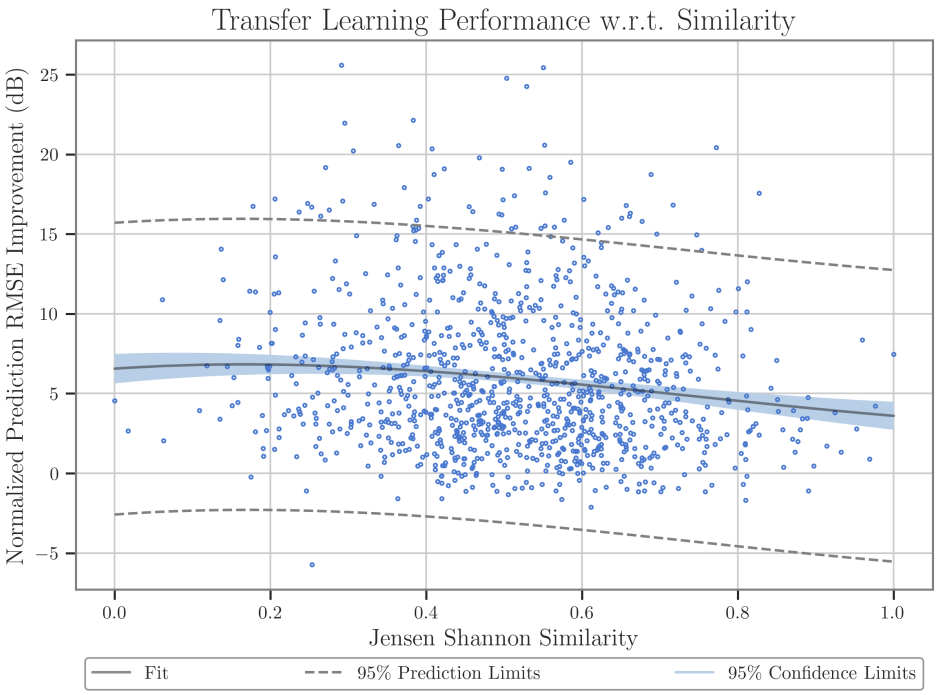


Figure 4.5: Source Cell Model RMSE improvement after Transfer Learning w.r.t Jensen Shannon similarity.

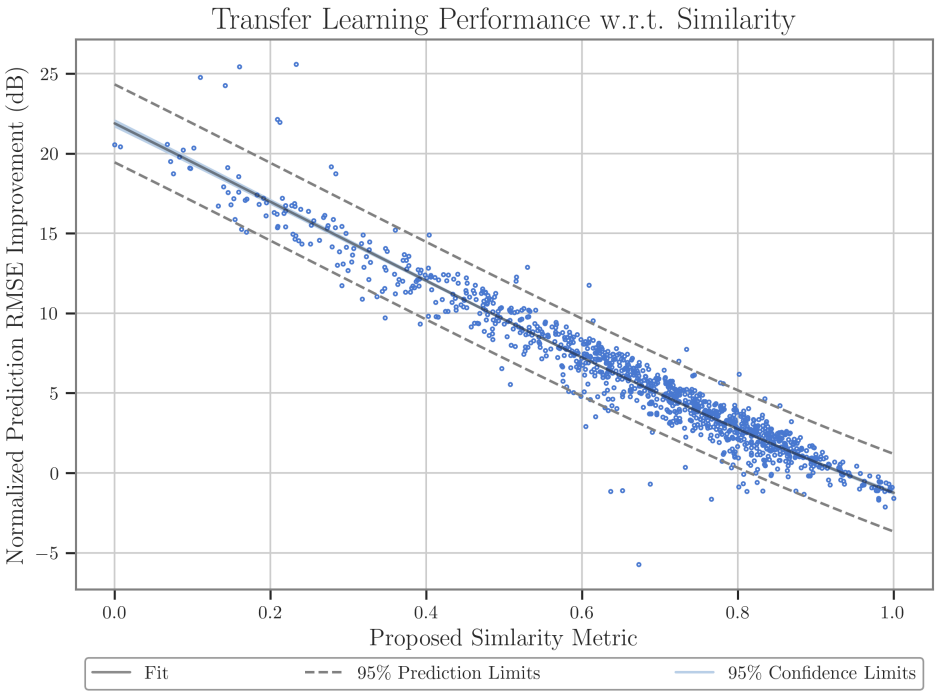


Figure 4.6: Source Cell Model RMSE improvement after Transfer Learning w.r.t proposed ML-based domain-aware similarity metric.

Results Discussion

The performance of the similarity metrics is evaluated by comparing the TL gain of each source to the target transfer with respect to the similarity of the corresponding metric as shown in Fig. 4.3 - Fig. 4.6. Each dot in these results shows the TL gain of a source to target transfer experiment, and overall 1089 experiments are carried out. The solid line in the center shows the regression fitting of all 1089 points, and the shaded region around this line shows the 95% confidence interval. Finally, the external dotted lines show the 95% prediction interval of the fitted regression curve.

All the aobe plots depicts the abovementioned aspects of TL gain with respect to the corresponding similarity metric, i.e., Wasserstein, Jensen Shannon, Energy Based and Proposed ML-based metric. In all these results, we aim to establish if there is any correlation between the TL gain and the respective similarity adopted to achieve that gain. Comparing all four results for this criteria, show that only the proposed ML-based similarity metric exhibits a clear linear trend between similarity and TL gain.

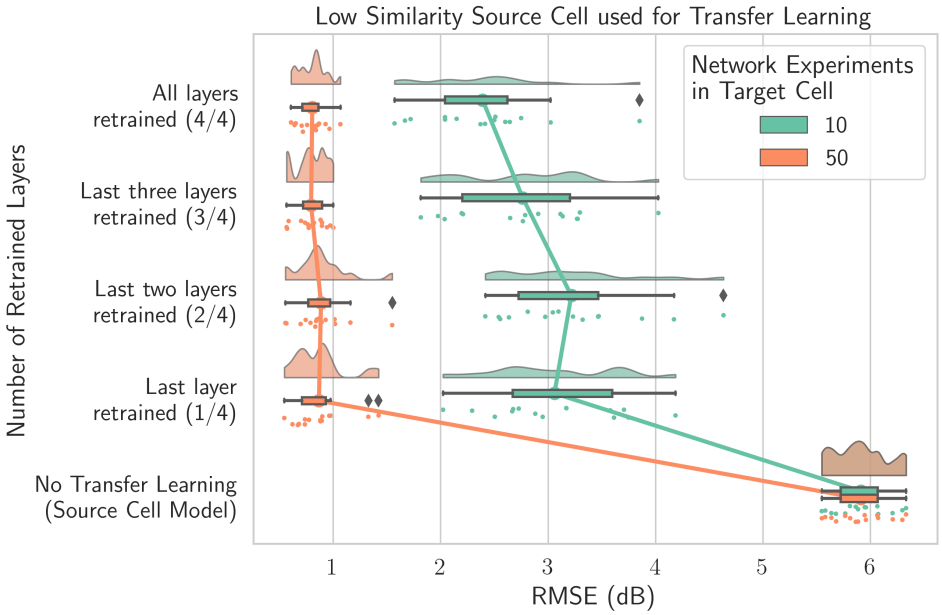


Figure 4.7: Low Similarity between Source and Target Cell. Prediction RMSE of the source model is reduced after transfer learning, i.e., *positive transfer*

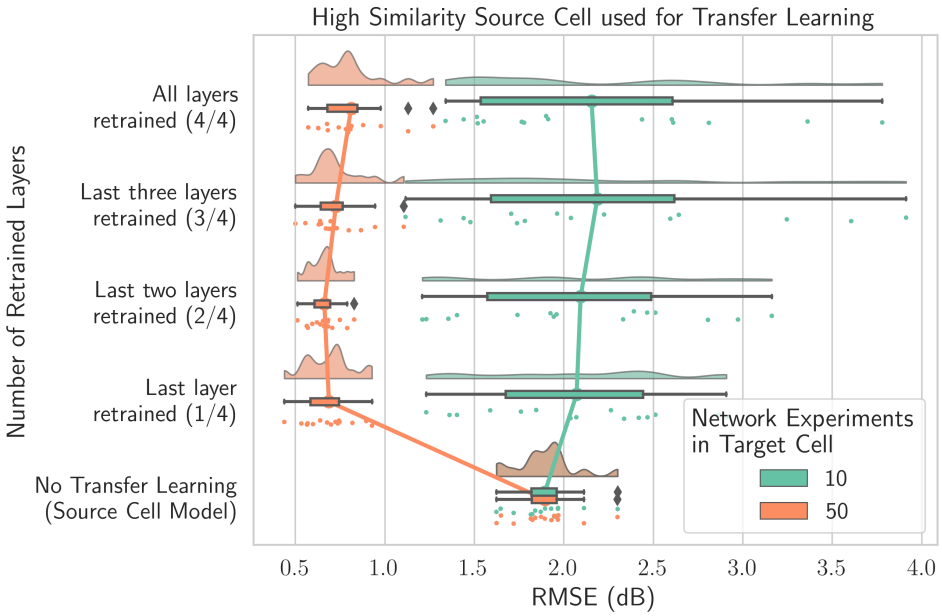


Figure 4.8: High Similarity between Source and Target Cell. Prediction RMSE of the source model is increased after transfer learning, i.e., *negative transfer*

4.4 Key Insight for Transfer Learning Strategy in a Sparse and Dynamic Network

This section discusses some insights learned by adopting the proposed ML-based similarity metric for selecting deep transfer learning strategies. The proposed similarity metric can be used as a guiding tool for the selection of the appropriate number of layers for retraining. For this purpose, we have designed a binary decision-based tree diagram shown in Fig. 4.9 where the recommendation for the TL strategy, described by the percentage of DNN layers to be retrained, is given by considering two cases of ML-based similarity between source and target cells, and two cases of target cell data representativeness using insights from Fig. 4.7 - Fig. 4.8. Specifically, when the similarity is high, and target cell data is representative, the number of retraining layers should be less than 25% of the total layers in the source cell DNN model (e.g., last layer out of 4 hidden layers should be re-trained). However, within the same high case of similarity, if the target cell data is not representative, there is no point/benefit of adopting the transfer learning, and the source cell model should be directly used. However, in the case of low similarity between

the source and target cell, it is observed from our experiments that all the layers should be re-trained to get the most transfer learning gain (or prediction RMSE improvement).

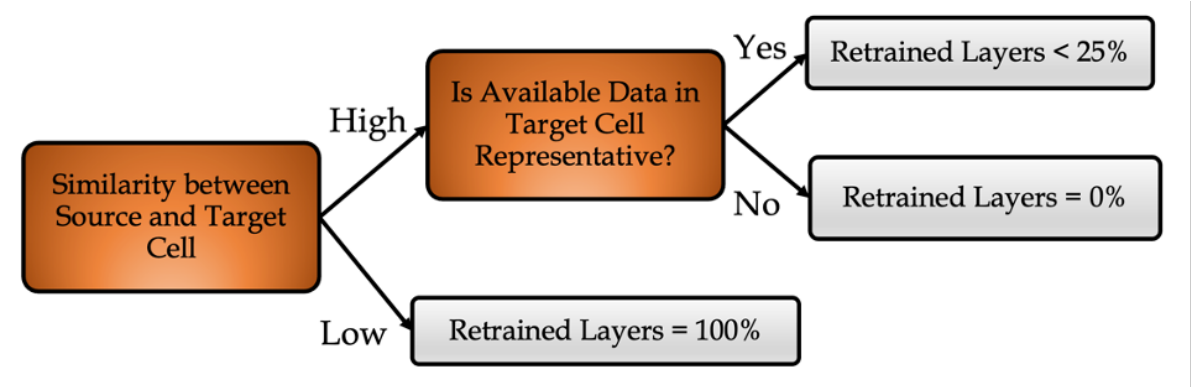


Figure 4.9: Transfer Learning Strategy in a Sparse and Dynamic Network

The representativeness of data is a subjective matter that varies based on the specific nature of the dataset under consideration. In our study, it was assumed that a count of 10 network experiments would not provide a representative dataset for the development of a network behavior prediction model. Conversely, by applying domain-specific knowledge, it was inferred that a count of 50 network experiments would yield a dataset sufficiently representative for modeling cell-level Key Performance Indicators (KPIs) by employing variables such as Antenna Tilt, Antenna Height, and Transmit Power. It should be noted that the adequacy of these counts will fluctuate based on the characteristics of the dataset and the model constructed from it, requiring meticulous analysis to ascertain the representativeness of a given set of measurements within a certain context. Future research may undertake a quantification of dataset representativeness, utilizing domain knowledge in a manner akin to our proposed framework for similarity quantification. Such investigations, however, fall outside the purview of the present dissertation and are left for subsequent studies.

4.5 Conclusion

In this chapter, a novel AI-based domain-aware cell similarity prediction model is proposed to address the challenge of selecting suitable transfer candidates (similar cells) in a dynamic and unrepresentative environment that is robust to the spatio-temporal dynamics in the network. By leveraging latent features such as geographical clutter pattern which can act as a proxy for spatial dynamics, and user mobility and traffic pattern etc., which can act as a proxy for temporal dynamics in the network, the proposed framework can measure similarity between cells deployed across different geographical environments and with different user traffic and mobility patterns. The proposed novel set of latent features can be computed for each cell using already collected network data (e.g., subscriber trace data, cell performance management KPI/counter data, geographical clutter data etc.). This data is readily available to the operators, without the need for further standardization. Furthermore, by doing extensive transfer learning experiments between different combinations of selected data rich cells, a 44-dimensional ML-based similarity metric is trained. This similarity metric takes as an input the existing network environment data of the source and target cell, and accurately predict the potential gain that can be achieved with transfer learning between two cells, before even collecting network experiment data in a new area. This can greatly help in Day 0 and Day 1 planning for deploying new macro and small cells and for optimizing existing sites. This is because using the proposed metric the operators can measure the similarity between the new area (target cell) and the existing pool of candidate source cells. It can help figure out which cell can most accurately represent the target cell environment for creating network behavior models through transfer learning, thus handling the spatial dynamicity challenge. Similarly, even if the underlying network conditions change over time, operators can accurately measure the cell similarity at any given time and fine-tune the model (using a new source cell based on updated network conditions), thus handling the temporal dynamicity challenge.

CHAPTER 5

Conclusion and Future Works

5.1 Conclusions

This dissertation outlines the application of Artificial Intelligence (AI) and Machine Learning (ML) methods in network design, offering promising solutions to the challenges inherent in mathematical models. The AI-based data-driven network models introduced can be used for system-level network planning and post-deployment optimization, effectively orchestrating a range of network parameters to optimize performance while reducing human intervention. Although network behavior models often have vast parameter spaces due to the complexity of emerging wireless networks, this dissertation leverages a domain knowledge-inspired, first-principles-based approach to make this problem manageable by AI.

However, the adoption and usability of AI/ML-based network models still face significant challenges. These include the integration of domain knowledge and model interpretability in ML-based network models, sparsity, and uneven distribution of real network training data collected by operators, and the dynamic nature of the underlying network environment.

The major contributions of this dissertation address these challenges by developing novel AI-based network behavior models. These models are domain-aware, interpretable, and robust to data sparsity and network dynamicity, offering increased efficiency and precision in network planning, design, and optimization. Three major contributions are outlined in this dissertation.

Firstly, an AI-based approach to radio propagation modeling was proposed to estimate the received signal strength. By employing a novel set of key predictors, the approach was

able to represent the physical and geometric structure of different propagation environments, mitigating the limitations of conventional models. The framework's robustness to data sparsity and its scalability, combined with the Light Gradient Boosting Machine's computational performance and predictive prowess, demonstrated strong performance. Additionally, interpretability, a common challenge in machine learning models, was addressed through SHApley Additive exPlanations (SHAP) method. This allowed for the generation of practical insights for the tuning of network configurations, selective enrichment of training data, and the building of lighter ML-based propagation models.

Secondly, a novel method for transfer learning was proposed, which presents a significant improvement in prediction accuracy compared to state-of-the-art DNN-based methods. It also offered the possibility of data reduction for equal performance, bringing forth potential savings for network operators. The conducted extensive transfer learning experiments under various data sparsity conditions provided insights into the design of an optimal transfer learning model based on available target domain data.

Finally, a domain-aware cell similarity prediction model was proposed, capable of handling the spatio-temporal dynamics in the network. By leveraging latent features, this model measured similarity between cells deployed in different geographical environments with varying user traffic and mobility patterns. This model, built on readily available data, could predict potential gains with transfer learning between two cells without additional data collection. This model will greatly aid in Day 0 and Day 1 planning for deploying new macro and small cells and optimizing existing sites. It has the potential to handle both spatial and temporal dynamicity challenges, thus introducing an innovative approach to network behavior models through transfer learning.

In conclusion, the contributions of this dissertation have addressed the key challenges that hinder the adoption of AI/ML-based network models, thus paving the way for their increased application in the field of network design and optimization. This work, by advancing AI-based network modeling, has illuminated pathways for future research and

practical application in this ever-evolving field.

5.2 Future Works

The system level pathloss prediction model presented in Chapter 2 combined with the interpretability analysis thus manifests a framework that can act as a cornerstone for the AI-based closed-loop automation solutions, as opposed to the 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 lass 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 the development of link-level and system-level channel models for higher frequency bands. This is especially needed for next-generation 3D heterogeneous cellular networks in mmWave/Terahertz bands with flying Unmanned Aerial Vehicle (UAV)-based BSs [60, 61] and Reconfigurable Intelligent Surfaces (RIS) [62], since the propagation conditions will be significantly different from the sub-6 GHz bands and conventional network planning using empirical model will cease to be a viable option.

An additional potential enhancement to the existing framework involves strengthening its resilience to variations in system parameters, including factors like Antenna Tilt Angle among others. This objective could be achieved through two plausible strategies. One approach is to augment the diversity of the training data, encompassing a variety of combinations of system parameters, thereby allowing the model to capture a broader range of parameter interactions and dependencies. Alternatively, the principles of transfer learning could be harnessed to facilitate the adaptation of the model from one set of

parameter configurations to another, thereby enhancing the model's versatility across different system settings.

Furthermore, our future research plans are focused on advancing the state of closed-loop automation solutions through the application of transfer reinforcement learning. We aim to build upon the novel findings proposed in this dissertation and extend them to more complex and challenging scenarios in telecommunication networks, particularly in ultra-dense millimeter-wave (mmWave) 5G New Radio (NR) heterogeneous networks (HetNets). Our forthcoming investigation will involve developing a Distributed Self-Coordinated Framework for Joint Optimization of Coverage, Capacity, Mobility Load Balancing, and Energy Saving. This framework will leverage the power of Multi-Agent Deep Transfer Reinforcement Learning to enable highly efficient and effective management of network resources. We believe that the application of transfer reinforcement learning in this context can significantly enhance the adaptability and performance of our network solutions. Through the process of transfer learning, our models will be able to apply the knowledge gained from a network environment to other related environments, thus accelerating the learning process and improving overall network performance.

Bibliography

[1] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE network*, vol. 28, no. 6, pp. 27–33, 2014.

[2] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas and propagation Magazine*, vol. 45, no. 3, pp. 51–82, 2003.

[3] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 255–270, 2012.

[4] *Urban Transmission Loss Models for Mobile Radio in the 900- and 1,800 MHz Bands: (revision 2)*, ser. COST 231: European Cooperation in the Field of Scientific and Technical Research, 1991.

[5] V. Erceg, K. Hari, M. Smith, D. Baum, K. Sheikh, C. Tappenden, J. Costa, C. Bushue, A. Sarajedini, R. Schwartz *et al.*, "Channel models for fixed wireless applications (IEEE802.16.3 c-01/29r4)," *Broadband Wireless Working Group, IEEE P802.16*, 2001.

[6] Forsk, *Model Calibration Guide*, AT340_MCG_E0, France, 2018. [Online]. Available: http://www.forsk.com

[7] 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.

[8] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2735–2747, 2010.

[9] M. Piacentini and F. Rinaldi, "Path loss prediction in urban environment using learning machines and dimensionality reduction techniques," *Computational Management Science*, vol. 8, no. 4, pp. 371–385, 2011.

[10] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on

machine learning: Principle, method, and data expansion," *Applied Sciences*, vol. 9, no. 9, p. 1908, 2019.

[11] M. Ribero, R. W. Heath, H. Vikalo, D. Chizhik, and R. A. Valenzuela, "Deep learning propagation models over irregular terrain," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4519–4523.

[12] S. P. Sotiroudis, 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 and Wireless Propagation Letters*, vol. 12, pp. 364–367, 2013.

[13] M. Ayadi, A. B. Zineb, and S. Tabbane, "A UHF path loss model using learning machine for heterogeneous networks," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 7, pp. 3675–3683, 2017.

[14] 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 Engineering*, vol. 5, no. 1, p. 1444345, 2018.

[15] 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. Advanced Int. Conf. Telecommun*, 2014, pp. 1–5.

[16] L. Dai, H. Zhang, and Y. Zhuang, "Propagation-model-free Coverage Evaluation via Machine Learning for Future 5G Networks," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1–5.

[17] A. Ghasemi, "Data-driven prediction of cellular networks coverage: An interpretable machine-learning model," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 604–608.

[18] "Forsk Atoll," https://www.forsk.com/atoll-overview/.

[19] "Aster Propagation Model," https://www.forsk.com/aster-propagation-model/.

[20] Forsk, *Aster Technical Reference Guide*, AS271_TRG_E0, France, 2019. [Online].

Available: http://www.forsk.com

[21] "EGS Technologies," http://www.egstech.com/.

[22] H. N. Qureshi and A. Imran, "Optimal bin width for autonomous coverage estimation using MDT reports in the presence of user positioning error," *IEEE Communications Letters*, vol. 23, no. 4, pp. 716–719, 2019.

[23] 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, 2020.

[24] 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 Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7538–7550, 2017.

[25] Y. Wang, K. Venugopal, A. F. Molisch, and R. W. Heath, "MmWave vehicle-to-infrastructure communication: Analysis of urban microcellular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7086–7100, 2018.

[26] N. Altman and M. Krzywinski, "Simple linear regression," 2015.

[27] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[28] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees. statistics/probability series," 1984.

[29] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[31] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[32] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, 2018, pp. 6638–6648.

[33] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[34] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[35] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, 1980.

[36] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on selected areas in communications*, vol. 17, no. 7, pp. 1205–1211, 1999.

[37] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[38] I. M. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Mathematical modelling and computational experiments*, vol. 1, no. 4, pp. 407–414, 1993.

[39] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1094–1111, 2018.

[40] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[41] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[42] "TreeSHAP," https://shap.readthedocs.io/en/latest/generated/shap.explainers.Tree.html.

[43] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on*

Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, conference Name: IEEE Transactions on Knowledge and Data Engineering.

[44] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer Learning for Wireless Networks: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 110, no. 8, pp. 1073–1115, 2022.

[45] J. Chuai, Z. Chen, G. Liu, X. Guo, X. Wang, X. Liu, C. Zhu, and F. Shen, "A Collaborative Learning Based Approach for Parameter Configuration of Cellular Networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 1396–1404, iSSN: 2641-9874.

[46] C. Parera, Q. Liao, I. Malanchini, C. Tatino, A. E. C. Redondi, and M. Cesana, "Transfer Learning for Tilt-Dependent Radio Map Prediction," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 829–843, 2020.

[47] C. Parera, A. E. Redondi, M. Cesana, Q. Liao, and I. Malanchini, "Anticipating Mobile Radio Networks Key Performance Indicators with Transfer Learning," in *2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*. IEEE, 2021, pp. 1–8.

[48] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: A Transfer Actor-Critic Learning Framework for Energy Saving in Cellular Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, Apr. 2014, conference Name: IEEE Transactions on Wireless Communications.

[49] C. Parera, A. E. C. Redondi, M. Cesana, Q. Liao, L. Ewe, and C. Tatino, "Transferring Knowledge for Tilt-Dependent Radio Map Prediction," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[50] M. S. Abrishami, A. E. Eshratifar, D. Eigen, Y. Wang, S. Nazarian, and M. Pedram, "Efficient Training of Deep Convolutional Neural Networks by Augmentation in Embedding Space," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*, Mar. 2020, pp. 347–351, iSSN: 1948-3287.

[51] W. Fang, C. Chen, B. Song, L. Wang, J. Zhou, and K. Q. Zhu, "Adapted Tree Boosting for Transfer Learning," *arXiv:2002.11982 [cs, stat]*, Apr. 2020, arXiv: 2002.11982. [Online]. Available: http://arxiv.org/abs/2002.11982

[52] F. Moradi, R. Stadler, and A. Johnsson, "Performance Prediction in Dynamic Clouds Using Transfer Learning," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 242–250.

[53] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[54] D. Pino, J. García, F. Fernández, and S. S. Vyetrenko, "Similarity metrics for transfer learning in financial markets," 2021.

[55] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[56] J. Carroll and K. Seppi, "Task similarity measures for transfer in reinforcement learning task libraries," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005, pp. 803–808 vol. 2.

[57] I. Pisa, A. Morell, J. L. Vicario, and R. Vilanova, "Transfer learning suitability metric for ann-based industrial controllers," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–8.

[58] H. Larsson, J. Taghia, F. Moradi, and A. Johnsson, "Source Selection in Transfer Learning for Improved Service Performance Predictions," in *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021, pp. 1–9.

[59] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-Free Renewable Scenario Generation Using Generative Adversarial Networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265–3275, May 2018.

[60] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang, "A survey on 5G millimeter wave communications for UAV-assisted wireless networks," *IEEE Access*, vol. 7, pp. 117 460–117 504, 2019.

[61] H. N. Qureshi and A. Imran, "On the Tradeoffs Between Coverage Radius, Altitude, and Beamwidth for Practical UAV Deployments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2805–2821, 2019.

[62] 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, 2019.