UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADDRESSING MOBILITY CHALLENGES IN AI-ENABLED EMERGING 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

MARVIN C. MANALASTAS
Norman, Oklahoma
2023

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/337708

ADDRESSING MOBILITY CHALLENGES IN AI-ENABLED EMERGING CELLULAR NETWORKS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Ali Imran, Chair

Dr. James J. Sluss, Jr.

Dr. Samuel Cheng

Dr. Timothy Ford

© Copyright by MARVIN C. MANALASTAS 2023
All Rights Reserved

Acknowledgments

I am deeply grateful to Dr. Ali Imran, my adviser throughout my Masters and PhD, for his invaluable support and guidance. His expertise, insights, and constructive feedback have been instrumental in shaping my research and in helping me achieve my academic goals. His unwavering dedication and passion for research have been truly inspiring and have helped me gain a deeper understanding of the significance of research. Thank you, Dr. Ali, for being an exceptional mentor and for your unwavering commitment to my academic and professional development.

My heartfelt appreciation also goes to Pres. Jim Sluss, Dr. Samuel Cheng, and Dr. Timothy Ford for taking the time to review my manuscript and attend my thesis defense. Their valuable feedback and suggestions were immensely helpful in improving my work.

I express my sincere appreciation to Denise Davis and Krista Pettersen for their exceptional support and assistance in handling all administrative tasks during my five years of study at OU-Tulsa. Their dedicated efforts and tireless work behind the scenes made my academic journey smooth and hassle-free.

I extend my sincere thanks to my friends and colleagues at AI4Networks Research Center, to Asad, Umar, Shahrukh, Usama, Sajid, Waseem, Tabish, Aneeqa, Dr. Haneya, Dr. Nabeel, and Dr. Fahd for their support and assistance during my academic journey. I feel proud and privileged to be a part of this Center, which provided me with a nurturing and stimulating environment to learn and grow.

I wish to express my utmost gratitude to the numerous teams and individuals whose concerted efforts brought about the successful development of SyntheticNET simulator and TurboRAN testbed, and made the research findings presented in Chapters 2 and 3 of this dissertation possible. Your dedication, expertise, and collaboration were integral to the realization of this project, and I am truly grateful

for the invaluable contributions each one of you made.

Finally, I wish to dedicate this work to my beloved family, including my *Nanay* (Ester) and *Tatay* (Cezar Jr.), my brothers (Mark and Jerwin), my sister (Donna) and my nephew (Mikhail). Their unwavering support, unconditional love, and endless encouragement have been the greatest source of strength and motivation throughout my academic journey. I am deeply grateful for their constant belief in me and for being my pillar of support.

Table of Contents

1 Introduction 1

1.1 Background and Motivation 1

1.2 Handover in Cellular Networks and the Select KPIs it Affects . . . 3

1.2.1 Handover Success Rate 4

1.2.2 Radio Link Failure Rate 5

1.2.3 Handover Signalling Overhead 5

1.2.4 Signal-to-Interference-plus-Noise Ratio 5

1.2.5 Throughput 6

1.3 Problem Statements and Outline of the Proposed Solutions 6

1.4 Current and Planned Dissemination and Publications 7

1.5 Organization 12

2 Design and Development of 3GPP Simulator for Investigating Mobility Related Problems and Solutions 13

2.1 Introduction 13

2.1.1 Related Work 14

2.1.2 Contributions 17

2.1.3 Chapter Organization 18

2.2 System-Level Simulators for 5G&B Networks 19

2.3 System-level Simulators Evaluation Leveraging the Proposed Metrics 24

2.3.1 New Insightful 5G-Specific Evaluation Metrics 24

2.3.2 Realism Evaluation 27

2.3.3 Completeness Evaluation 29

2.3.4 Computational Efficiency Evaluation 29

2.3.5 Evaluation Summary and Simulator Applicability 31

2.4 Paving the Way Towards 6G: Future Requirements and Challenges of the Next Generation Simulators 34

2.4.1 Extreme Computational Complexity in 6G 36

2.4.2 Computational Demanding Aspects of Running Simulations in Future Networks 37

2.4.3 Proposed Architecture to Address the Computational Complexity of Next Generation Network Simulators 40
2.4.4 Case Study: Efficient Generation of Synthetic Network Data for Training AI Models 47
2.5 Conclusion 52

3 Design and Development of Testbed for Realistic Mobility Evaluation . 54
3.1 Introduction 54
3.1.1 Contributions and Organization 55
3.2 Testbeds for 5G and Beyond Cellular Networks 56
3.3 TurboRAN Overview, Design, and Planning Considerations 58
3.3.1 TurboRAN Overview 60
3.3.2 Network Design and Planning 62
3.3.3 Antenna Selection and Remote Electrical Tilt Configuration 68
3.3.4 Power Amplifiers and Low Noise Amplifiers 72
3.3.5 Protection Cabinet and Cooling System 74
3.3.6 Miscellaneous TurboRAN Components 78
3.4 Case Study: Mobility-based Inter-frequency Handover 78
3.5 Conclusion 84

4 Addressing Key Challenges in Using AI for Mobility Management and Zero Touch Automation in Emerging Cellular Networks 86
4.1 Introduction 86
4.2 AI as Key Enabler for Zero-Touch Optimization in Emerging Cellular Networks 88
4.3 The Two Key Challenges that Debilitate the Use of Off-the-shelf AI in Cellular Networks 88
4.3.1 Sparse data from a live network 89
4.3.2 Hyperparameter optimization 90
4.4 Addressing the Key Challenges in AI for Emerging Cellular Networks 91
4.4.1 Addressing the data sparsity challenge 91
4.4.2 Addressing the hyperparameter optimization challenge . . . 94
4.5 AZTEK: A Framework for Zero-Touch Optimization in Emerging Cellular Networks 97

4.6 A Case Study on AI-based Zero-Touch Optimization in Emerging Cellular Networks . . . 100
4.7 Conclusion . . . 102

5 An ML-based Framework Towards Improving the Mobility Robustness in Emerging Cellular Networks . . . 104
5.1 Introduction . . . 104
5.1.1 Relevant Work . . . 105
5.1.2 Proposed Approach and Contributions . . . 107
5.2 System Model and Data Generation . . . 111
5.2.1 Inter-frequency Handover Success Rate . . . 111
5.2.2 Factors Affecting Handover Success Rate . . . 112
5.2.3 Simulation Setup and Data Generation . . . 114
5.3 Data Augmentation and Evaluation of Synthetic Data Quality . . . 117
5.3.1 Effect of Imbalance Data in HOF Prediction . . . 117
5.3.2 Data Augmentation Techniques to Address Class Imbalance 118
5.3.3 Data Augmentation Leveraging the Borderline Concept . . . 119
5.3.4 Evaluation Methods for Synthetic Data . . . 120
5.4 AI-Based Handover Failure Prediction Model . . . 121
5.4.1 HOF Prediction Models . . . 122
5.4.2 Domain Knowledge Based Feature Engineering . . . 124
5.5 Handover Failure Mitigation Algorithm and Performance Evaluation 126
5.5.1 Tx Power Tuning based HOSR Improvement Algorithm . . . 126
5.5.2 Handover Failure Mitigation Scheme Evaluation . . . 128
5.6 Results and Discussion . . . 130
5.6.1 Synthetic Data Evaluation Results . . . 130
5.6.2 Handover Failure Prediction Results . . . 133
5.6.3 Shift in the Decision Boundary after Class Balancing . . . 136
5.6.4 Handover Failure Mitigation Results . . . 136
5.7 Conclusion . . . 139

6 Introduction of a Novel Handover Parameter for Emerging AI-enabled Cellular Networks . . . 141

6.1 Introduction . . . 141

6.1.1 Related Work . . . 142

6.1.2 Contributions . . . 144

6.2 UIO Introduction . . . 145

6.2.1 Limitations of CIO . . . 149

6.2.2 UIO Compliance to 3GPP Standard Handover . . . 151

6.2.3 UIO Optimization Methods . . . 152

6.3 Framework and System Model . . . 155

6.3.1 Network-Originating UIO Generation Framework . . . 156

6.3.2 Problem Formulation . . . 157

6.3.3 Simulation Setup and Data Generation . . . 160

6.4 Impact of UIO on KPIs . . . 164

6.4.1 Impact of UIO in SINR, HOSR, and SigO . . . 164

6.4.2 Multi-objective Optimization . . . 165

6.5 ML-based with Hueristics UIO Generation . . . 166

6.5.1 Performance Comparison of Machine Learning Models . . . 166

6.5.2 Objective Function Optimization . . . 167

6.6 Conclusion . . . 169

7 Conclusions and Future Work . . . 170

References . . . 172

List of Figures

1.1 The current prevailing KPI optimization highly involves human interventions and hit-and-trial approach. 4

1.2 High-level overview of the challenges in mobility management in AI-enabled cellular network and the proposed solutions 8

2.3 List of evaluated simulators and comparison metrics in relevant literature. 17

2.4 Iron triangle of cellular network simulator. 25

3.7 Number of handovers under mobile user settings. 83

3.8 Number of handovers under static user settings. 84

4.1 A Venn diagram showing the complex relationship between COPs and KPIs. 89

4.2 Off-the-shelf ML/DL techniques with corresponding hyperparameters. 91

4.3 AZTEK: Proposed zero-touch optimization framework addressing key challenges on the use of AI in mobile networks. 98

4.4 The optimal performance of ML models is achieved by addressing the sparsity and hyperparameter optimization challenge. 101

4.5 Comparison in the performance between brute force approach and Genetic Algorithm. 102

5.2 Impact of user speed in occurrences of HOFs. 114

5.3 Network deployment layout and RSRP plot for simulations. 115

5.4 Confusion matrix of the classifiers trained using imbalanced data set. 118

5.5 Sobol sensitivity analysis of the input features used for HOF prediction. 125

5.6 PDF comparison between real and synthetic data using different data augmentation techniques. 131

5.7 Best and worst pairwise correlation difference comparison. 132

5.8 AUROC (a) and AUPRC (b) comparison between different augmentation techniques using different AI algorithms. 132

5.9 Time complexity comparison of different AI models. 135

5.11 Handover failure mitigation performance using different data augmentation techniques. 137

5.12 HOSR ξ improvement comparison between algorithm used in [1] and proposed HOF mitigation algorithm. 139

6.2 Impact of CIO on static users. 150

6.3 HO message flow with the incorporation of the proposed UIO. . . . 152

6.4 Proposed UIO generation using ML with heuristics approach. . . . 156

6.5 Network deployment and RSRP traces. 161

6.6 Heatmap illustrating the influence of a user's optimal UIO (UIO_o) on other users' SINR (a) and SigO (b) 163

6.7 Variations on the optimal UIO settings for different KPIs and user behavior. 165

6.8 Performance comparison between machine learning with heuristic optimization approach and brute force methods, while also examining the impact of varying the weights. 168

6.9 Performance comparison between CIO-optimized and UIO-optimized networks. 169

List of Tables

2.1 Summary of Comparison Between System-level 5G Simulators using Traditional Metrics. 21

2.2 Proposed 5G-specific Evaluation Metrics for System-level 5G Simulators. 26

2.3 Summary of Realism Comparison Between Different 5G System-level Simulators 28

2.4 Summary of 5G Features Supported by Different 5G System-level Simulators. 29

2.5 Summary of the Methods Applied to Improve the Computational Efficiency of Different 5G System-level Simulators 30

2.6 Evaluation of the readiness of 5G system-level simulators to act as DT kernel based on insightful 5G-specific metrics defined in Table 2.2 31

2.7 Applicability of different 5G system-level simulators to different research topics. 33

2.8 Summary of some of the most computationally demanding aspects when running a network simulator, how it affects simulation run time in simulating 5G network and the future implications on the upcoming 6G networks. 38

2.9 Network simulation settings. 49

3.1 5G network testbeds around the world. 59

3.2 A summary of cellular system research areas supported by TurboRAN. 60

3.3 ISM frequency bands supported by TurboRAN. 62

3.4 Comparison between select 5G network testbed solutions. 68

3.5 Key features of Amarisoft CallBox. 69

3.6 Selected antenna specification. 70

3.7 Power amplifiers and low noise amplifiers requirements. 73

3.8 Comparison of specific applications of enclosures for outdoor nonhazardous locations (from NEMA 250-2003). 75

3.9 Outdoor enclosure specification. 76

3.10 Maximum heat output of the equipment inside the cabinet. 78

3.11 Summary of intra-RAT measurement events for LTE and 5G NR. . 82

5.1 Description of Simulation Parameters 115
5.2 Hyperparameters Used for Training CTGAN 119
5.3 Different Set of Input Features and Their Effect in HOF Prediction 125
5.4 Summary of Evaluation Comparison Between Different Data Augmentation Techniques 132
5.5 Failed Compensation and False Mitigation Score Υ for Different values of α 138

6.1 Description of Simulation Parameters 162
6.2 Normalized RMSE comparison of different machine learning algorithms for SINR, HOSR, and SigO prediction. 167

Abstract

Ability to support seamless user mobility has been the Raison D'etre of mobile networks. In recent years, the cellular network industry has evolved significantly, and mobility management is becoming a major challenge. This is due to factors such as densified deployment, stringent Quality of Service (QoS) requirements of diverse use cases like next-generation URLLC, and evolved network architecture. In this thesis, we identify and tackle the pressing challenges of mobility management in emerging cellular networks, which if not addressed, can potentially become the network's Achilles' heel. We address three primary challenges: lack of suitable tools for investigating mobility, radio link failures due to handover failures, and limitations of the existing handover parameters in the standards for mobility management optimization.

To address first challenge, we propose a tri-pronged approach: developing a computationally efficient and realistic mobility simulator, designing and deploying a state-of-the-art experimental testbed called TurboRAN, and introducing AZTEK, an AI-enabled handover parameter optimization framework that can work with limited training data generated either from the simulator, testbed or real network.

To address the second challenge, we propose and evaluate TORIS (Transmit Power Tuning-based Handover Success Rate Improvement Scheme), a novel data-driven solution to reduce inter-frequency handover failures. TORIS consists of an AI-based model to predict handover failures and a heuristic scheme for tuning the transmit power of cells. Unlike conventional methods, TORIS proactively adjusts cell power when a handover failure is anticipated.

To address the third challenge, we propose and analyze a novel parameter called user individual offset (UIO), which considers user-specific behaviors (i.e., speed, direction, service requirement) for value selection. Our results show that UIO can

help resolve long-standing challenges in mobility management without tradeoffs between key performance indicators.

CHAPTER 1

Introduction

1.1 Background and Motivation

The exponential growth of mobile data traffic due to the increasing number of capacity-hungry devices has resulted in a demand for emerging cellular networks to provide massive coverage and capacity. While physical layer techniques such as massive MIMO, enhanced frame structure, and advanced channel coding have been developed to address this issue, network densification is considered the most effective solution [2]. However, the deployment of a large number of base stations of different types operating in a wide range of frequencies has created new challenges in network management [3].

Mobility or handover management is one aspect that is significantly impacted by the deployment of dense base stations. This is because the number of handovers is directly proportional to the number of base stations and inversely proportional to the coverage size. Deploying dense small cells can result in an enormous number of handovers, leading to various handover-related issues such as early, delayed, wrong, and ping-pong handovers, and ultimately, Radio Link Failures (RLFs) [4]. These issues can cause degradation in several key performance indicators, including retainability, throughput, latency, and increased signaling overhead, thereby reducing the overall user quality of experience and quality of service. Therefore, cellular networks must be equipped with efficient mobility management systems to avoid KPI degradation [5].

Designing and maintaining an efficient mobility management system is a complex task that involves several factors. One crucial element is the availability of platforms

that enable solution validation, as testing them on a live network is impractical. These platforms can include computer-aided numerical simulations, simulators, and testbeds. However, even with these tools, achieving efficient mobility management is challenging if it relies heavily on human skills.

One critical aspect that requires attention is the optimal setting of mobility-related configuration and optimization parameters, which greatly impact handover performance [6]. However, finding the ideal parameter setting is challenging, especially in current 5G networks where dozens of handover parameters must be adjusted to achieve the desired handover performance. The current industry practice of optimizing mobility heavily relies on the deep domain knowledge and experience of engineers. However, the strong dependency on human effort makes this approach prone to errors and time-consuming, to say the least. In the last decade, some level of automation has been realized in the form of Self Organizing Network (SON) [7]. However, the state-of-the-art (SOTA) SON solutions are not capable to deliver the desired zero-touch automation due to the lack of intelligence and their reactive nature.

In incoming 6G networks, this number is expected to increase due to the increased complexity of the system, making it nearly impossible to discover the optimal handover parameter setting. Fortunately, recent advances in artificial intelligence (AI) have opened up new opportunities for achieving zero-touch network automation, where the network can autonomously make decisions based on real-time data and adjust network parameters dynamically to meet user demands. This approach has the potential to revolutionize mobility management and enable the efficient operation of complex networks without human intervention.

While AI can assist in enabling zero-touch mobility optimization, the current nature of mobility parameter setting hinders the full utilization of its potential. To reduce the probability of handover failures, operators can use AI to fine-tune handover-

related parameters such as handover margin, time-to-trigger, and cell individual offset. However, despite their usefulness, the inflexibility of these parameters still poses several limitations. For instance, configuring cell individual offset per neighbor relation has limited influence over the specific user equipment that will be impacted by CIO adjustment. Thus, modifying the CIO relationship between neighboring cells may lead to favorable outcomes for some users but adversely affect others.

1.2 Handover in Cellular Networks and the Select KPIs it Affects

Handover (HO) is the mechanism used to transfer a connected user equipment (UE) from one base station to another. As the UE moves away from the serving base station, also known as the source cell, and towards the next base station, the receive signal level and condition from the source cell decrease while they increase for the target base station. To ensure that the UE maintains uninterrupted service, the source and target cells must coordinate with each other to facilitate the handover process.

The 3GPP TS 36.331 [8] defines standard events that aid handover decision. These events include seven events in 3GPP Release 8 (A1, A2, A3, A4, A5, B1, and B2), with an additional Event A6 introduced in Release 10. The first five events are typically used for intra-system handover (e.g., 5G to 5G or LTE to LTE), while B1 and B2 are used for inter-RAT handover (e.g., 5G to LTE or 3G). Event A6 is used for the handover of a secondary cell in Carrier Aggregation (CA). Of the listed events, events A3, A4, and A5 are the most commonly used in evaluating handover decisions. These events involve measuring the neighboring cell and evaluating whether the condition is better than the serving cell or meets a certain threshold. By using these events with the right parameter settings, UE will always camp on the base station with the best signal condition.

While the evolution of cellular network technology is fast paced, the way these these

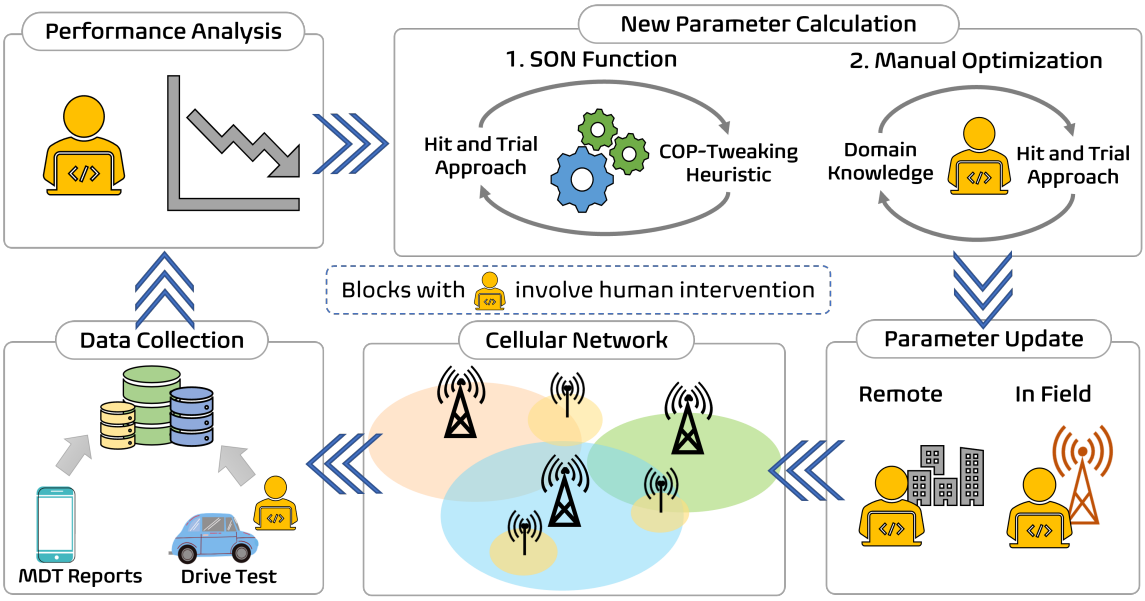


Fig. 1.1: The current prevailing KPI optimization highly involves human interventions and hit-and-trial approach.

handover parameters are set lags. For instance, the recently deployed 5G network still relies on a manual approach to tune these HO configuration and optimization parameters (COPs) to optimize key performance indicators (KPIs) as shown in Fig. 1.1. Ensuring optimal setting of these handover parameters is one of the primary goals of network operators, as it directly impacts several KPIs of the network, including:

1.2.1 Handover Success Rate

Handover Success Rate (HOSR) is a key performance indicator (KPI) that directly measures the performance of handover in a network. It is divided into two categories: Intra-frequency HOSR, which evaluates the success rate of handovers occurring within the same layer, and Inter-frequency HOSR, which evaluates the success rate of handovers occurring between different layers. Poor radio conditions, high interference, high traffic in the target cell, and problematic neighboring relations, such as handover to overshooting cells, are some of the common causes of poor HOSR.

1.2.2 Radio Link Failure Rate

When a user equipment (UE) is in connected mode, it is said to be in RRC Connected State. However, under certain conditions, the RRC connection can fail, and the user declares a RLF. This occurs when the signal strength deteriorates below a certain level, making it difficult for the UE to maintain a good connection to the serving base station. RLF can happen for various reasons, but the most common is due to handover failures. When handover from the source to the target cell fails, the UE enters a poor SINR area. The longer the handover takes, the longer the UE remains in the poor SINR area, increasing the likelihood of RLF occurrence.

1.2.3 Handover Signalling Overhead

Signaling overhead refers to the amount of signaling messages that are exchanged between the UE and the base station, to maintain a connection during handover. Signaling overhead can consume network resources and reduce the overall capacity of the network. It can also increase the delay in delivering user data and reduce the quality of service (QoS) for users.

1.2.4 Signal-to-Interference-plus-Noise Ratio

SINR is a metric that quantifies the quality of a received signal. Although it is not a standardized measurement in 3GPP specifications, it is commonly used by UE and equipment vendors. Since it is not a reported measurement, the UE converts SINR to a Channel Quality Indicator (CQI), which provides an estimation of the signal condition. Poor handover performance can result in delayed handovers, which, in turn, can cause the UE to remain connected to a cell with a poor SINR.

1.2.5 Throughput

Throughput refers to the rate at which data packets are transmitted over the air interface. This rate is primarily influenced by two factors: the amount of bandwidth allocated to the transmission (expressed in terms of PRBs, or Physical Resource Blocks) and the SINR (Signal-to-Interference-plus-Noise Ratio) of the transmission. Since SINR is closely tied to handover performance, the effectiveness of handovers can have a significant impact on throughput rates.

1.3 Problem Statements and Outline of the Proposed Solutions

The above discussions highlights the importance of optimizing handover procedures to ensure a smooth transition from one base station to another, minimizing the impact on critical KPIs. Although various solutions exist to enhance these KPIs and establish efficient mobility management strategies, none of them comprehensively addresses the wide range of challenges involved [1, 9–27]. Thus, in this dissertation, we identify and address the mobility challenges in AI-enabled emerging cellular networks. We present the problems (**P**) we identified, as well as our proposed solutions (**S**) for each of these challenges as shown in Fig. 1.2.

P1: Investigating mobility in emerging cellular networks is challenging due to the complexity introduced by various factors such as handover parameter interplay, user trajectory, user speed, varying network coverage, and traffic patterns. Analytical models are not suitable for mobility investigation due to this complexity, while mobility simulators require system level details that are often missing in most simulators. Additionally, most existing testbeds lack multi-cell over-the-air deployment necessary for mobility modeling.

To address these challenges, we propose a tri-pronged approach. First, we develop a simulator capable of modeling mobility realistically by using a novel architec-

ture that achieves realism and completeness without compromising computational efficiency (**S1-A**). Second, we design, develop, and deploy a testbed that has an over-the-air interface and allows mobility-related experiments (**S1-B**). Finally, to maximize the utility of the simulator or testbed-generated synthetic data, we propose a framework to address machine learning (ML) model training data sparsity (**S1-C**).

P2: Radio link failures (RLF) caused by handover failures are becoming more pressing in emerging cellular networks due to network densification, multi-band operation, and the rising complexity of handover parameter optimization. While various solutions have been proposed to optimize handover in emerging networks, our investigations show that the current standard has limitations that do not allow for optimal mobility management.

To address this challenge, we propose and evaluate an AI-based solution to optimize handover in emerging networks by predicting and mitigating handover failures in advance (**S2**).

P3: Optimization of parameters (e.g., using CIO or TTT) in existing standards cannot meet the stringent mobility requirements of 6G and often leads to improvements in certain KPIs at the cost of other KPIs.

To overcome this challenge, we propose a novel parameter called user individual offset (UIO) (for which we file an invention disclosure) that, according to our results, can help solve age-old challenges in mobility management (e.g., CIO-based solutions that constantly demand tradeoffs between KPIs) without tradeoffs (**S3**).

1.4 Current and Planned Dissemination and Publications

Awards

A1. Awarded Gallogly College of Engineering Dissertation Excellence Award by

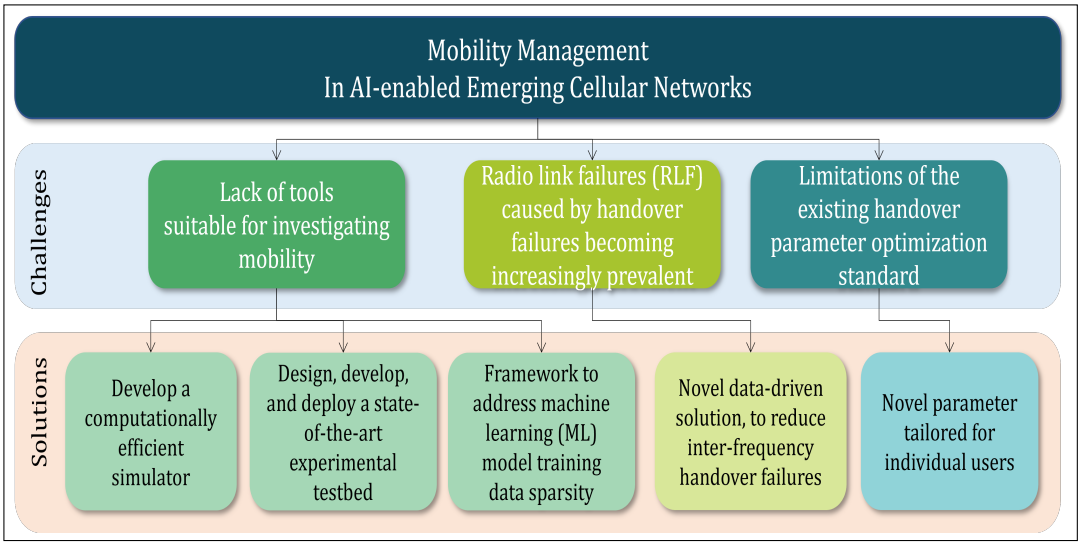


Fig. 1.2: High-level overview of the challenges in mobility management in AI-enabled cellular network and the proposed solutions

the University of Oklahoma, Spring 2023.

A2. Won the best research presentation award for academic year 2019-2020 in ECE/TCOM department at OU Graduate Research Meeting.

Peer-Reviewed Journal Articles

J1. A. Ijaz, W. Reza, H. Farooq, **M. Manalastas**, and A. Imran, "An AI-Enabled Framework to Defend Ingenious MDT-based Attacks on the Emerging Zero Touch Cellular Networks," in IEEE Network, 2023.

J2. H. N. Qureshi, **M. Manalastas**, M. U. B. Farooq, A. Imran, Y. Liu and M. O. A. Kalaa, "5G-Enabled Healthcare in Mobile Scenarios: Challenges and Implementation Considerations," in IEEE Network, doi: 10.1109/MNET.125.2200243.

J3. S. M. A. Zaidi, **M. Manalastas**, M. U. B. Farooq, H. Qureshi, A. Abu-Dayya and A. Imran, "A Data Driven Framework for QoE-Aware Intelligent EN-DC Activation," in IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 2381-2394, Feb. 2023, doi: 10.1109/TVT.2022.3211741.

J4. M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, A. Abu-Dayya and A. Imran, "A Data-Driven Framework for Inter-Frequency Handover Failure Prediction and

Mitigation," in IEEE Transactions on Vehicular Technology, vol. 71, no. 6, pp. 6158-6172, June 2022, doi: 10.1109/TVT.2022.3157802.

J5. M. Umar Bin Farooq, **M. Manalastas** et al., "A Data-Driven Self-Optimization Solution for Inter-Frequency Mobility Parameters in Emerging Networks," in IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 2, pp. 570-583, June 2022, doi: 10.1109/TCCN.2022.3152510.

J6. M. Manalastas et al., "Design Considerations and Deployment Challenges for TurboRAN 5G and Beyond Testbed," in IEEE Access, vol. 10, pp. 39810-39824, 2022, doi: 10.1109/ACCESS.2022.3166947.

J7. Qureshi, H.N.; **Manalastas, M.**; Ijaz, A.; Imran, A.; Liu, Y.; Al Kalaa, M.O. Communication Requirements in 5G-Enabled Healthcare Applications: Review and Considerations. Healthcare 2022, 10, 293. https://doi.org/10.3390/

J8. H. N. Qureshi, **M. Manalastas**, A. Imran and M. O. Al Kalaa, "Service Level Agreements for 5G-Enabled Healthcare Systems: Challenges and Considerations," in IEEE Network, vol. 36, no. 1, pp. 181-188, January/February 2022, doi: 10.1109/MNET.011.2100343.

J9. H. N. Qureshi, **M. Manalastas**, S. M. A. Zaidi, A. Imran and M. O. Al Kalaa, "Service Level Agreements for 5G and Beyond: Overview, Challenges and Enablers of 5G-Healthcare Systems," in IEEE Access, vol. 9, pp. 1044-1061, 2021, doi: 10.1109/ACCESS.2020.3046927.

J10. S. M. A. Zaidi, **M. Manalastas**, H. Farooq and A. Imran, "Mobility Management in Emerging Ultra-Dense Cellular Networks: A Survey, Outlook, and Future Research Directions," in IEEE Access, vol. 8, pp. 183505-183533, 2020, doi: 10.1109/ACCESS.2020.3027258.

J11. S. M. A. Zaidi, **M. Manalastas**, H. Farooq and A. Imran, "SyntheticNET: A 3GPP Compliant Simulator for AI Enabled 5G and Beyond," in IEEE Access, vol.

8, pp. 82938-82950, 2020, doi: 10.1109/ACCESS.2020.2991959.

J12. M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, and A. Imran, "Towards the Development of 6G System Level Simulators: Addressing the Computational Complexity Challenge" IEEE Wireless Communications Magazine [**submitted paper under review**]

J13. M. Manalastas, M. U. B. Farooq, 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 Wireless Communications Magazine. [**submitted paper under review**]

J14. M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, and A. Imran, "5G & Beyond Simulators: A Survey, Development Challenges, and Future Requirements" IEEE Communications Surveys and Tutorials [**under review**]

J15. M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, and A. Imran, "User Individual Offset (UIO): A Novel Handover Parameter for Next-Generation Cellular Networks" [**under review**]

Peer-Reviewed Conference Papers

C1. M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, and A. Imran, "Machine Learning-Based Handover Failure Prediction Model for Handover Success Rate Improvement in 5G", 2023 IEEE Annual Consumer Communications & Networking Conference (CCNC).

C2. S. M. A. Zaidi, **M. Manalastas**, and A. Imran, "An Analytical Model to Quantify the Effect of Handover and Cell Density on SINR in Emerging Cellular Networks", 2023 IEEE Annual Consumer Communications & Networking Conference (CCNC).

C3. M. U. Bin Farooq, **M. Manalastas**, H. Qureshi, Y. Liu, A. Imran and M. O. Al Kalaa, "MDT-based Intelligent Route Selection for 5G-Enabled Connected

Ambulances," 2022 IEEE International Conference on E-health Networking, Application & Services (HealthCom), Genoa, Italy, 2022, pp. 81-87, doi: 10.1109/HealthCom54947.2022.9982778.

C4. M. U. B. Farooq, **M. Manalastas**, S. M. A. Zaidi, A. Abu-Dayya and A. Imran, "Machine Learning Aided Holistic Handover Optimization for Emerging Networks," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 710-715, doi: 10.1109/ICC45855.2022.9839024.

C5. M. Nabeel, **M. Manalastas**, A. Ijaz, H. Refai and A. Imran, "Investigating Handover Behavior with 5G and Beyond TurboRAN Testbed," 2022 Seventh International Conference On Mobile And Secure Services (MobiSecServ), Gainesville, FL, USA, 2022, pp. 1-6, doi: 10.1109/MobiSecServ50855.2022.9727204.

C6. M. U. Bin Farooq, **M. Manalastas** et al., "Data Driven Optimization of Inter-Frequency Mobility Parameters for Emerging Multi-band Networks," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9348101.

C7. S. M. Asad Zaidi, **M. Manalastas**, A. Abu-Dayya and A. Imran, "AI-Assisted RLF Avoidance for Smart EN-DC Activation," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6

C8. 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), Odessa, Ukraine, 2020, pp. 1-6, doi: 10.1109/BlackSeaCom48709.2020.9235020.

C9. M. Manalastas, H. Farooq, S. M. Asad Zaidi and A. Imran, "Where to Go Next?: A Realistic Evaluation of AI-Assisted Mobility Predictors for HetNets," 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC),

Las Vegas, NV, USA, 2020, pp. 1-6, doi: 10.1109/CCNC46108.2020.9045127.

1.5 Organization

The dissertation is organized as follows: Chapter 2 presents the design and development of a 3GPP simulator for investigating mobility-related problems and solutions. It also addresses the challenges involved in developing advanced network simulators for mobility simulations and proposes solutions to overcome the computational complexity challenge. Chapter 3 focuses on the design and development of a testbed for realistic mobility evaluation. Chapter 4 proposes a framework for addressing key challenges in using AI for mobility management and zero touch automation in emerging cellular networks. Chapter 5 presents an ML-based framework aimed at improving the mobility robustness in emerging cellular networks. Chapter 6 introduces a novel handover parameter for emerging AI-enabled cellular networks. Finally, Chapter 7 presents the conclusion and outlines potential future work.

CHAPTER 2

Design and Development of 3GPP Simulator for Investigating Mobility Related Problems and Solutions

2.1 Introduction

System level 5G&B simulators are software applications that simulate the system-level behavior of a 5G&B network. These simulators are intended to mimic the complete 5G network architecture, including the radio access network (RAN) and user equipment (UE). 5G system-level simulators can be used to investigate and optimize the performance of 5G networks under different conditions and scenarios. In a controlled environment, they enable network designers and engineers to evaluate various deployment strategies, network configurations, and traffic scenarios.

Recent trends suggest that to provide a more advanced method of network modelling, the cellular network industry is moving towards digital twins (DTs), which are virtual and up-to-date representations of a physical system (i.e., cellular network) [28]. DTs are composed of three unique components: the physical product, the digital/virtual model of the product, and the interconnections represented by the data that reflects the current state of a live system, as shown in Fig. 2.1. While DTs are becoming more popular, three challenges limit their utility in the research and development of emerging cellular networks. First, DTs replicate a functioning network, limiting their applicability to technologies that have already been implemented rather than emerging networks that are not yet physically present. Second, for DTs to work effectively, they need real data and constant feedback from a real network [29]. However, unlike other domains where data is rich, data in mobile networks tends to be sparse [30]. Third, DTs lack generality since they are deployment

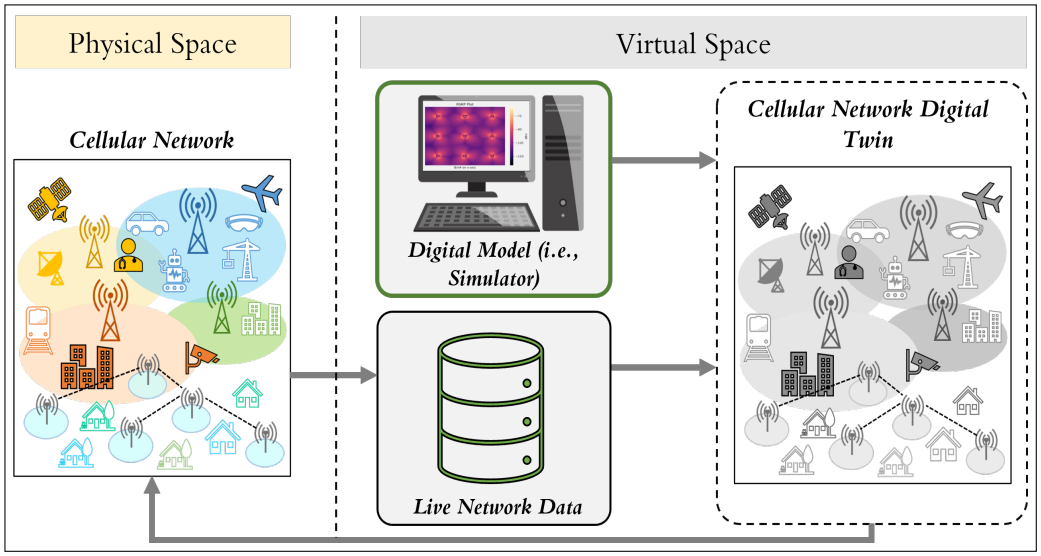


Fig. 2.1: Simulator utilized as digital twin kernel.

specific (i.e., DT for one location will not be suitable for another). Accordingly, it is critical to understand the capabilities of simulators and the attributes they must possess to facilitate the fruition of DT in emerging networks. In this chapter, on top of assessing these simulators, identifying their strengths and drawbacks, we also evaluate them based on a novel set of criteria to establish their suitability as digital twin kernels. These criteria include realism, completeness and computational efficiency. Using these criteria, this chapter provides insight into the present status of simulators to break this iron triangle and become DT-ready. Moreover, it is vital to raise awareness about the challenges and potential solutions in developing these simulators as well as the future requirements of simulators for the next generation cellular networks.

2.1.1 Related Work

Wireless networks (WN) have had a colossal impact on various sectors of human society, including communication, education, defense, security, healthcare, agriculture, and manufacturing, among others. It is a broad area of research and can be broken down into several sub-topics as shown in Fig. 2.2. These specific domains

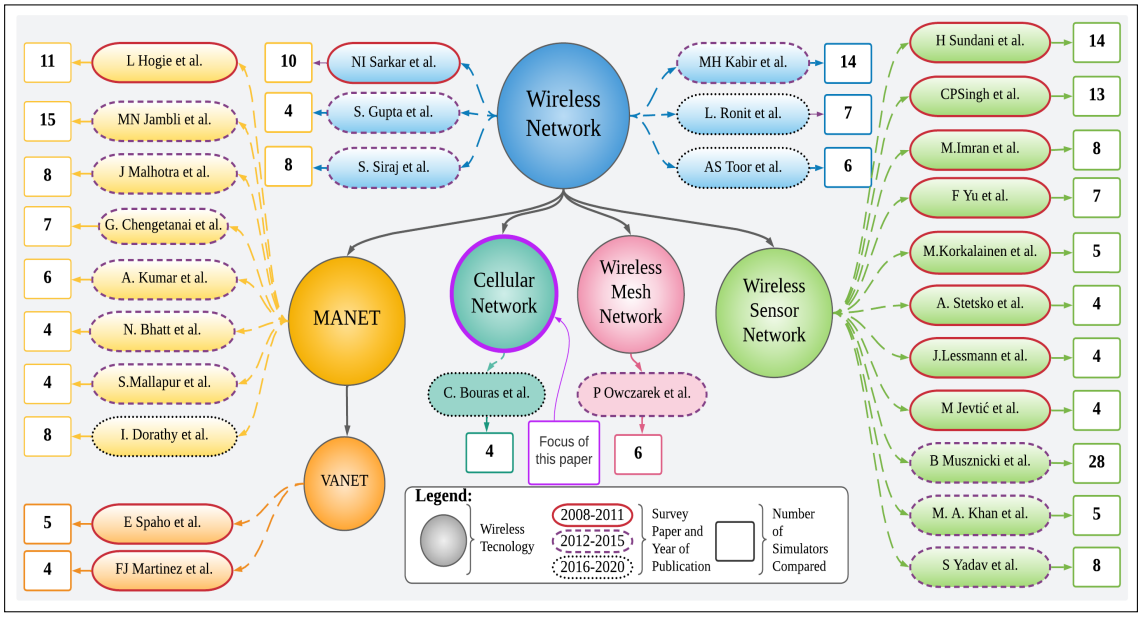


Fig. 2.2: Summary of available literature on simulator comparison in wireless networks. Inside the circles are the different domains of wireless networks. Inside the connected ellipses are the survey papers available in literature. The outline of these ellipses represents the year of publication. The figures inside the square represents the number of simulators included in each survey paper.

include Wireless Sensor Network (WSN), Wireless Mesh Network (WMN), cellular networks, Mobile Ad-hoc Network (MANET), and Vehicular Ad-hoc Network (VANET), which is a sub-category of MANET. The overwhelming demand to advance wireless communication has led to the development of several simulators that perform a broad array of functionalities.

There are several comparative studies on simulators for WNs in the literature. Fig. 2.2 shows a summary of the related literature we assimilate in this chapter. Currently, several surveys and comparative studies are focused on WN in general [31–36]. Meanwhile, other works also exist highlighting the simulators for specific sub-categories of WN. For instance, [37–43] are surveys on simulators used for MANETs, [44, 45] are works dedicated for VANETs simulators, while [46] confined the simulator discussions to WMN. Moreover, bulk of the papers on simulator survey and comparative studies are concentrated on WSN with 11 papers as of this writing [47–57]. Although some survey papers extensively cover a large number

of simulators, such as [31, 33, 38, 42, 50–52], evaluating 10 or more simulators, it is worth noting that many of these papers may be considered outdated since they were published between 2008 and 2015.

Unlike other branches of WN, such as WSN, WMN and MANETS, review papers dedicated on simulators pertinent to emerging cellular networks are scarce, as shown in Fig. 2.2. A review paper that is most related to the scope of this chapter is [58], which offers a comparative study on simulators specific to 4G and 5G cellular networks. However, this review paper has very limited scope as it considers only four simulators.

While there are other review papers available on cellular network simulators, such as those in [59–62], they primarily concentrate on the specifications of 5G simulators rather than their evaluation. These papers discuss various requirements for simulators, such as reusability, scalability, flexibility, multiple levels of abstraction, parallel processing, and the integration of link-, system-, and network-level simulators. However, it is worth noting that most of these requirements are not exclusive to 5G networks, and they provide only a cursory assessment of the challenges in developing 5G simulators. While [62] discusses the challenges associated with developing 5G simulators, the study is not comprehensive and does not offer insights into overcoming the challenges inherent in developing 5G&B simulators.

Along with the aforementioned articles on cellular network simulators, there exist publications that delve into simulator development [63–65]. In such works, authors not only expound on the features and capabilities of their respective simulators, but also perform comparative analyses with other 5G simulators, highlighting the strengths and advantages of their own tools. For instance, the authors in [64] compared their simulator with ten others with respect to supported features such as massive multi-input multi-output (MIMO), flexible numerology, random access procedure, mmWave propagation, and general information such as programming

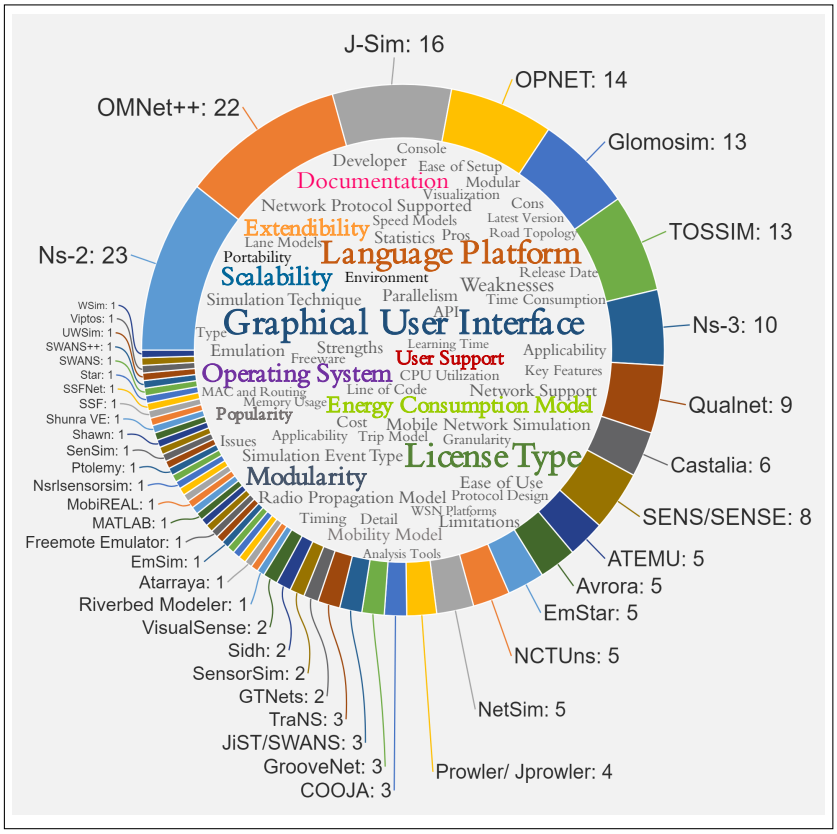


Fig. 2.3: List of evaluated simulators and comparison metrics in relevant literature.

language and license type. Similarly, authors in [63] developed a simulator and compared it to existing simulators using almost similar metrics as [64] with the inclusion of cloud computing capabilities. By discussing its primary features, the authors in [65] compared the simulator they developed with six other simulators.

2.1.2 Contributions

Examination of the available literature on WN simulators reveals that, in comparison to other domains, cellular network simulators lag in the availability of review papers. Meanwhile, those that are now available are either insufficiently comprehensive, have limited discussion of simulator development challenges and potential solutions, or do not use 5G-specific evaluation measures. Given the shortcomings of the current relevant works on cellular network simulators, there is a dire need to focus on 5G&B network simulators. In summary, the main contributions of this

work are as follows:

1. This chapter provides a comprehensive survey of the current simulators for 5G&B networks. In contrast to current literature, which concentrates on general-purpose simulators (shown in Fig. 2.3), this survey covers simulators targeted for specific 5G use cases. Our discussion includes over 17 system-level simulators. Comparative examination of different simulators will expedite and simplify the process of selecting an appropriate simulator.

2. This chapter present a new and insightful 5G&B network-specific evaluation metric for determining the status, fidelity level of implementation, and assumptions of each simulator. This metric measures the readiness of the simulators to act as kernels DTs. While traditional and generic metrics involve comparison in terms of graphical user interface, language platform, license type, modularity, etc., as shown in Fig. 2.3, the new metric revolves around assessing three major criteria: realism, completeness, and computational efficiency. This metric is then used to examine the capabilities and applicability of each simulator in different fields of research in 5G&B. The presented analysis can assist researchers in selecting the proper simulation tool for their needs.

3. This chapter highlight the looming challenge of extreme computational complexity of developing 6G-specific simulators. This chapter also identify the computationally complex aspects of simulating a 6G network and present a novel architecture to make simulators computationally efficient.

2.1.3 Chapter Organization

The structure of this chapter is organized as follows: Section 2.1 provides the introduction and relevant work. Section 2.2 presents a brief discussion of around 17

system-level simulators that can be leveraged by both academia and industry alike for research, network planning, and optimization. Section 2.3 presents the new insightful metrics for simulator evaluation based on three factors namely realism, completeness, and computational complexity. Section 2.4, extends the discussion future requirements and challenges of the next generation simulators. This section also present an architecture that enhances the computational efficiency of simulators, and provide a use case to demonstrate its impact. Finally, Section 2.5 concludes this chapter.

2.2 System-Level Simulators for 5G&B Networks

A typical cellular network system contains several interconnecting linkages between users and base stations, which are difficult to represent in detail without significantly increasing computational complexity. To minimize the computational cost, system-level simulators employ an abstracted link-level simulation [66]. Although link-level simulators are excellent at simulating the PHY layer of a cellular network, they are ineffective at simulating other layers, such as the MAC layer. Thus, system-level simulators are responsible for the detailed modeling of MAC layer functions, as well as various other algorithm designs, such as handover, interference management, admission control, link adaption, power control, and resource scheduling and allocation [67].

The discussion of system-level simulators is divided into two parts. This section begins with a discussion of specialized system-level simulators followed by the general system-level simulators.

Specialized system-level simulators

- **5GPy** [65]: 5GPy is a python-based, event-driven simulator developed as a

platform to perform C-RAN architecture simulations. 5GPy supports both small- and large-scale simulations for C-RAN.

- **C-RAN Simulator** [68]: This simulator is designed to provide a platform for system-level evaluation of 5G cloud-based networks. Some of the features implemented in C-RAN Simulator include centralized user scheduling, joint transmission of edge-users, and support for global per-antenna carrier aggregation.

- **Telco Cloud Simulator** [69]: Telco cloud simulator (TCS) is developed to simulate a virtualized 5G network and serves as a platform for conducting 5G cloud-related research.

- **V. Diaz et al.** [70]: This specialized simulator is a web-based path loss simulation tool based on 3GPP channel models for 5G network. Specifically, this simulator implements channel models standardized in 3GPP technical report TR 38.901 [71].

- **Py5cheSim** [72]: Py5cheSim is a Python-based, open-source simulator designed to model cell capacity in 5G&B networks and serves as the first simulator that enables network slicing at the RAN.

Generic system-level simulators

- **5G-air-simulator** [64]: 5G-air-simulator models various critical components designed for the 5G air interface at the system level. These key technical components include massive MIMO, extended multicast, and broadcast transmission schemes, predictor antennas, enhanced random access procedure, and NB-IoT. Additionally, it includes a variety of network architectures capable of simulating multiple cells and users, a wide selection of mobility models, and a high-fidelity link-to-system model for the physical and data-link layers.

Table 2.1: Summary of Comparison Between System-level 5G Simulators using Traditional Metrics.

Simulator	Simulator Type	User Interface	Language/ Platform	Open Source	License Type	Documentation and User Support	Modular
5GPy [65]	Specialized	Not specified	Python	Yes	General Public License	Limited	Yes
C-RAN Simulator [68]	Specialized	Command Line Interface	MATLAB	Not specified	Academic Use License	Limited	Yes
NYUSIM [73]	Specialized	Command Line Interface	MATLAB	Yes	Academic Use License	Extensive	Not specified
Telco Cloud Simulator [69]	Specialized	Graphical User Interface	Java	Not specified	Not specified	Limited	Not specified
V. Diaz et al. [70]	Specialized	Menu Driven Interface	PHP	No	General Public License	Limited	Not specified
Py5cheSim [72]	Specialized	Not specified	Python	Yes	General Public License	Limited	Yes
5G-air-simulator [64]	Generic	Command Line Interface	C++	Yes	General Public License	Extensive	Yes
5G K-simsys [74]	Generic	Graphical User Interface	C++	Yes	Academic Use License	Extensive	Yes
I. Belikaidis et al. [75]	Generic	Graphical User Interface	Not specified	Not specified	Not specified	Limited	Yes
M. Liu et al. [76]	Generic	Graphical User Interface	Not specified	Not specified	Not specified	Limited	Not specified
S. Cho et al. [62]	Generic	Not specified	C++	Not specified	Not specified	Limited	Yes
K. Bąkowski et al. [77]	Generic	Graphical User Interface	Not specified	Not specified	Not specified	Limited	Not specified
SiMoNe [10]	Generic	Graphical User Interface	Not specified	Not specified	Not specified	Limited	Yes
SyntheticNET [63]	Generic	Command Line Interface	Python	No	General Public License	Limited	Yes
Vienna 5G SL Simulator [78]	Generic	Command Line Interface	MATLAB	Yes	Academic Use License	Extensive	Yes
WiSE [11]	Generic	Graphical User Interface	C++	No	Not specified	Limited	Not specified
X. Wang et al. [79]	Generic	Not specified	Unknown	Not specified	Not specified	Limited	Not specified

- **5G K-simsys** [73]: 5G K-simsys is a modular and flexible system-level simulator. 5G K-simsys offers reuse capability for the majority of the modules and provides an easy-to-learn simulation environment.

- **I. Belikaidis et al.** [74]: This system-level simulator aims to develop a tool for simulating multi-connectivity scenarios. The authors utilized multiple channel models, traffic models, and mobility models for testing different service requirements of 5G networks.

- **K. Bąkowski et al.** [75]: The authors presented a system-level simulation tool

for evaluating some of the major research directions of the 5G network. These research directions include long-term interference mitigation, D2D resource allocation, and two-way relaying.

- **M. Liu et al.** [76]: The authors of this simulator introduced a novel design for 5G system-level simulations. Since 5G was still non-existent during the time of its publication, the authors applied the proposed approach to a heterogeneous environment comprised of LTE-A and IEEE 802.11-based Wi-Fi.

- **S. Cho et al.** [62]: This study demonstrated a work-in-progress simulator that addressed several of the challenges associated with 5G system-level simulators, including scalability, reusability, accuracy in the presence of abstraction, computational efficiency, and tight coupling between link and system-level simulations.

- **SiMoNe** [10]: Simulator for mobile networks also known as SiMoNe, is a system-level simulator developed primarily to facilitate the testing with novel self-organizing networks (SON) solutions. Through seamless parameter variations, SiMoNe provides a platform for testing a large variety of network configurations.

- **SyntheticNET** [63]: SyntheticNET is a cellular network simulator built in Python for 4G and 5G&B networks in compliance with 3GPP Release-15. It is a modular, flexible and versatile simulator supporting advanced features like adaptive numerology, handover, and futuristic database-aided edge computing to name a few.

- **Vienna 5G system-level simulator** [77]: Vienna system-level simulator is built in MATLAB and is one of the most popular simulators in the field of mobile network communications. Vienna system-level simulator allows large-scale multi-tier network performance evaluation and supports various types

of network nodes. A modular and flexible architecture combined with efficient object-oriented programming (OOP)-based implementation has enabled Vienna system-level simulator to support large-scale simulations.

- **WiSE** [11]: Wireless Simulator Evolution (WiSE) is a powerful system-level simulator originally developed for 4G environments. Recently, the scope is extended to support 5G by adding vital aspects of 5G NR such as new channel models, and 5G NR RAT features including scalable numerologies, flexible duplex, and code block group (CBG)-based transmission.

- **X.Wang et al.** [78]: The authors proposed a new design for a system-level simulator specifically for heterogeneous networks. This simulator decoupled the radio resource heads (RRHs) from eNodeB by assigning a cell ID to each RRH. This enabled a more comprehensive testing and analysis of algorithms such as load balancing, intra-site coordinated beamforming, and scheduling.

In Table 2.1, the different system-level simulators is evaluated based on general information such as user interface type, language, and traditional metrics such as documentation, user support, and modularity. The evaluation shows that the majority of system-level simulators employ a graphical user interface. Meanwhile, the most frequently used programming language or platform for developing these simulators is C++, followed by MATLAB. SyntheticNET and 5Gpy are Python-based simulators while TCS and V. Diaz et al. have leveraged Java and PHP, respectively. While the majority of simulators have limited documentation and user support, 5G-air-simulator, 5G K-SimSys, and Vienna 5G all have detailed documentation and user guides. Additionally, the majority of simulators are modular in design. Meanwhile, the licensing type for a large number of simulators is not indicated in the referenced manuscripts.

2.3 System-level Simulators Evaluation Leveraging the Proposed Metrics

This section, evaluates the system-level simulators to highlight their respective strengths and limitations as well as their readiness to act as DT kernel. It begins by proposing novel insightful 5G-specific evaluation metrics and then conducting a full evaluation of simulators using this metrics. Finally, this section assess each system-level simulator's applicability to relevant research topics.

2.3.1 New Insightful 5G-Specific Evaluation Metrics

While traditional metrics previously shown in Fig. 2.3 can be used to evaluate 5G simulators on a basic level, more insightful 5G-specific metrics can provide a more complete picture of the capabilities, features, and depth that are crucial throughout the simulator selection process. In this backdrop, we propose a novel set of metrics that revolve around three core characteristics of a simulator: realism, completeness, and computational efficiency.

1. Realism: We define realism as the accurate representation of cellular network properties such as propagation, user mobility patterns, network elements, network layout, and handover. This aspect is essential for establishing whether simulators are DT-ready, since it directly gauges the network resemblance of the digital model. However, simulator realism is diminished when models are oversimplified and inappropriate assumptions are made during the simulator's development. Among these simplifications are the assumption of perfect hexagonal network geometry, a perfect omni-directional antenna pattern, and considering only static users. Often, this oversimplistic approach is motivated by the need for a computationally efficient simulator or by a lack of in-depth domain knowledge. Without realistic implementation, a simulator cannot fully develop into an effective tool.

2. Completeness: The second metric, simulator completeness, is measured by evalu-

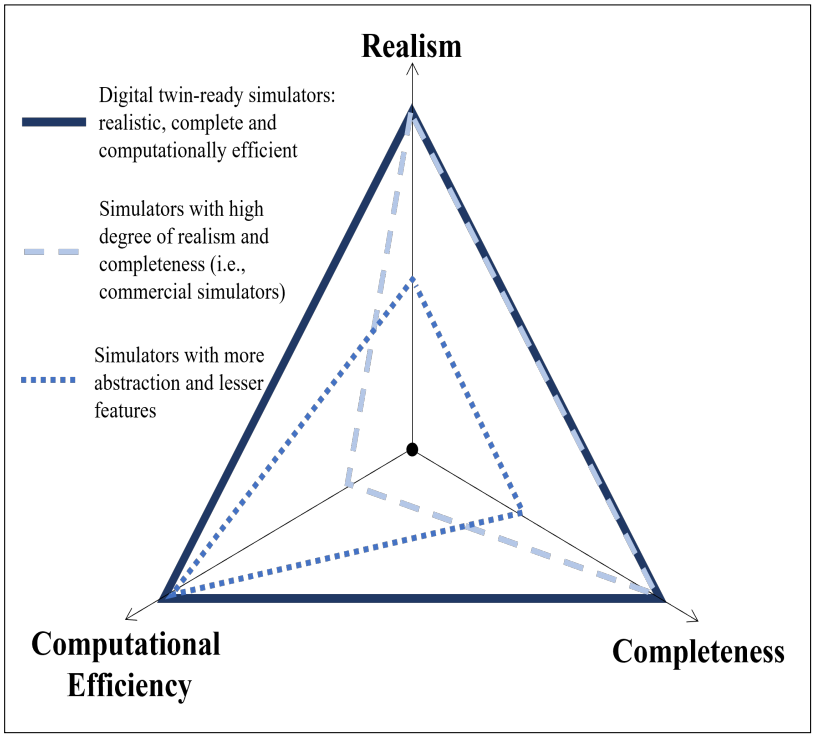


Fig. 2.4: Iron triangle of cellular network simulator.

ating the 5G features and functions implemented within a simulator. Completeness assesses the extent to which the simulator covers all relevant aspects of the system being simulated. We highlight eight of the most critical components of a 5G network, including mmWave operation, scalable subcarrier spacing, massive MIMO, beamforming, multi-RAT dual connection, carrier aggregation, dynamic TDD, and capacity components as part of completeness assessment. The completeness of a simulator quantifies the number of functions it can execute as a DT kernel.

3. Computational Efficiency: Finally, a simulator, although realistic and complete, still falls short as DT-ready without computational efficiency. Generally, the simulator's computational efficiency decreases with the increase in the number of network elements, i.e., base stations, users, types of services, number of parameters tested, etc. We evaluate the third metric, the computational efficiency of the system-level simulators, in terms of the strategies implemented to reduce the computational complexity of simulations.

Table 2.2: Proposed 5G-specific Evaluation Metrics for System-level 5G Simulators.

	5G-Specific Metrics	3-star rating (★★★)	2-star rating (★★)	1-star rating (★)
REALISM	Propagation Model	Models pathloss, shadowing, small scale fading, penetration loss, realistic antenna pattern, buildings, blockages and vegetation.	Similar to 3-star rating but lacks the modeling of buildings, blockages and vegetation.	Models limited propagation characteristic such as pathloss and shadowing.
	Channel Model and Calibration	Models variety of stochastic 5G channels with 3GPP or ray-tracing based calibrations.	Models variety of stochastic 5G channels without advanced calibrations.	Lacks comprehensive modeling of stochastic 5G models.
	Handover Model	Detailed 3GPP based handover modeling.	Simple handover models lacking 3GPP based handover modeling.	No handover model.
	User Mobility Model	Supports a wide variety of realistic mobility models along with the ability to import real mobility traces.	Lacks the support of realistic mobility models and cannot import real mobility traces.	Does not support mobile users.
	Traffic Demand Model	Contains multiple traffic models to support different use cases.	Single or fixed traffic model.	No traffic model.
	Network Geometry	Supports multiple network geometries and customizable BS deployment.	Supports only multiple network geometries.	Supports a single network geometry deployment.
COMPLETENESS	» mmwave » Scalable Numerology » Massive MIMO » Beamforming » Dual Connectivity » Carrier Aggregation » Dynamic TDD » Bandwidth Parts	Supports at least half of the listed 5G NR features.	Supports only 1 of the listed 5G NR features.	No support for any of the listed 5G NR features.
COMPUTATIONAL EFFICIENCY	Incorporates methods to reduce the computational complexity and improve running time of simulations.	Employs several techniques to improve the computational efficiency and running time.	Employs a single technique to improve the computational efficiency and running time.	Does not employ any technique to improve the computational efficiency and running time.

Fig. 2.4 the shows the devised metric in a form of an iron triangle. In this context, a DT-ready simulator should possess high degree of each of the three criteria. Table 2.2 provides a detailed description of each metric that is used to evaluate the readiness of the simulators to act as DT kernels. To provide a clear evaluation of the simulators, they are rated on a scale of 1 to 3 stars, with 3 being the highest rating. A star rating is assigned to each category based on the degree of achievement in meeting the criteria for realism, completeness, and computational

efficiency. In order to be considered DT-ready, a simulator must exhibit a high degree of performance in each of these areas. This rating system provides a quick and easy way to compare the performance of different simulators and identify areas that need improvement. In the next discussions we evaluate several system-level simulators based on the proposed metrics.

2.3.2 Realism Evaluation

Table 2.3 presents the comparison of system-level simulators based on their degree of realism. The evaluation shows that Vienna SL 5G Simulator and WiSE have the most practical propagation models. Both simulators incorporate the impact of blockages into their propagation models with WiSE even considering the effect of oxygen absorption. 5G-air-simulator, 5G K-SimSys, Vienna SL 5G Simulator, and WiSE are among the simulators with a rich selection of standard channel models. However, while the other three simulators specified performing channel model calibrations, Vienna SL 5G Simulator has not specified any calibrations in their channel models. Meanwhile, SiMoNe is the only simulator that implements the ray tracing propagation approach.

With regards to handover modeling, only two simulators, SyntheticNet and SiMoNe, explicitly discuss the implementation of handover functions. While some may include handover support, it is not clearly stated in the accessible manuscripts for these simulators. Similarly, SyntheticNet and SiMoNe have the most comprehensive mobility models including the ability to import real mobility traces. Meanwhile, 5G-air-simulator, *M.Liu et al.*, and SiMoNe have the most diverse traffic models. Additionally, SiMoNe also leverages a real traffic database consisting of video streaming, web browsing, and voice traffic.

Vienna SL 5G Simulator is by far the most extensive simulator for network geometry while both SyntheticNet and SiMoNe support importing of user-defined topology.

Table 2.3: Summary of Realism Comparison Between Different 5G System-level Simulators

System-level Simulators	Propagation	Channel Model and Calibration	Handover Model	User Mobility Model	Traffic Demand Model	Network Geometry
5G-air-simulator [64]	Path loss, shadowing, penetration loss, and fast fading	UMa, SMa, RMa, UMi, UFe, WINNER DL, Basic DL and IMT models (Calibrated based on industry model)	None	Standstill, RWP, linear movement, random walk, and Manhattan	Video streaming, voice traffic, web browsing, and constant bitrate	Hexagonal grid
5G K-SimSys [73]	Not specified	InH, Dense urban, UMa, and RMa (Calibrated based on industry models)	None	Not specified	Full buffer, and non-full buffer model	Hexagonal grid
I.Belikaidis et al. [74]	Not specified	Not specified	None	Random walk, and linear motion	FTP traffic, full buffer and HTTP traffic model	Not specified
K.Bakowski et al. [75]	Large-scale fading, and shadowing	O2O, O2I, and I2I	None	Shortest path, and random movement following the street	Not specified	Madrid grid model (MGM)
M.Liu et al. [76]	Path loss, shadowing, antenna gain, and penetration loss	UMa, UMi, RMa, and SMa	None	Random walk, standstill, and fix-track	FTP traffic model, full buffer traffic model, and HTTP traffic model	Not specified
S. Cho et al. [62]	Not specified	Not specified	None	Not specified	Not specified	Hexagonal grid
SiMoNe [10]	Antenna pattern, and building data	Macro-predictor, 3D ray-tracer for outdoor cells, and analytical 3D ray-launcher	3GPP X2-based handover	RW, real trajectories, customized 3D, RWP, vehicular, pedestrian and cyclist, and 3D indoor	Real traffic database (video streaming, web browsing, and voice traffic)	User-defined topology, and hexagonal grid
Synthetic-NET [63]	Shadowing, realistic antenna pattern, and influence of blockage objects	FSPL, 2-slope pathloss, and other empirical models	3GPP X2-based handover	RWP, manhattan, SLAW, traces from mobility generator (SUMO, BonnMotion), and real traces	Not specified	User-defined topology
Vienna SL 5G Simulator [77]	Path loss, shadow fading, antenna pattern, small scale fading, and influence of blockage objects	Fixed, FSPL, InH, RMa, SMa, UMa, 3D-UMa, UMi, and 3D-Umi	None	RWP, and pre-defined trajectories	Not specified	Gauss cluster, hexagonal grid, hexagonal ring, manhattan grid, predefined, uniform cluster, and uniform Piosson Point Process
Wise [11]	Influence of blockage objects, and oxygen absorption effects	InH, Dense Urban, RMa, UMa, and 3D-UMa (Calibrated based on 3GPP calibration campaign)	None	Not specified	Full buffer	Hexagonal grid
X. Wang et al. [78]	Antenna pattern, and shadowing	ITU-R Pedestrian-B	None	Not specified	Not specified	Hexagonal grid

2.3.3 Completeness Evaluation

The comparison of the completeness of several system-level simulators is shown Table 2.4. It is worth noting that none of the currently available simulators incorporates all the listed key features of 5G NR. While the majority of simulators offer scalable numerology, massive MIMO, and beamforming, the implementation of mmWave propagation remains scarce despite being one of the cornerstones of 5G&B networks. In fact, only three simulators, 5G K-simsys, Vienna SL 5G Simulator and WiSE, support mmWave. Furthermore, support for multi-RAT dual connectivity, carrier aggregation, and dynamic TDD are not commonly implemented. Lastly, none of the current simulators support the simulation of bandwidth parts. This completeness analysis demonstrates that current simulators still fall short of fully implementing 5G NR features.

Table 2.4: Summary of 5G Features Supported by Different 5G System-level Simulators.

Simulator	mmWave	Scalable Numerology	Massive MIMO	Beamforming	Dual Connectivity	Carrier Aggregation	Dynamic TDD	Bandwidth Parts
5G-air-simulator [64]		✓	✓	✓		✓	✓	
5G K-simsys [73]	✓	✓	✓	✓	✓			
I.Belikaidis et al. [74]			✓	✓		✓		
K.Bakowski et al. [75]								
M. Liu et al. [76]					✓			
S. Cho et al. [62]								
SiMoNe [10]					✓			
SyntheticNET [63]		✓			✓	✓	✓	
Vienna SL 5G Simulator [77]	✓	✓	✓	✓				
Wise [11]	✓	✓	✓	✓			✓	
X. Wang et al. [78]								

2.3.4 Computational Efficiency Evaluation

Several strategies for speeding up the simulation process for 5G networks are now being employed in 5G system-level simulators, as shown in Table 2.5. For instance, *S. Cho et al.*, SyntheticNET, and Vienna 5G SL Simulator all exploit parallel processing to accelerate simulation run-time. Meanwhile, SyntheticNET, and Vienna 5G SL Simulator include pre-generated reference signal received power (RSRP) and

Table 2.5: Summary of the Methods Applied to Improve the Computational Efficiency of Different 5G System-level Simulators

System-level Simulators	Methods Applied to Improve Computational Efficiency
5G-air-simulator [64]	Object-oriented programming (OOP)-based implementation
5G K-simsys [73]	None
I. Belikaidis et al. [74]	None
K.Bakowski et al. [75]	None
M.Liu et al. [76]	None
S. Cho et al. [62]	Parallel processing
SiMoNe [10]	Uplink abstraction based on [79]
SyntheticNET [63]	1. Parallel Processing 2. Channel traces pre-generation 3. User traces pre-generation
Vienna SL 5G Simulator [77]	1. Abstractions of PHY and MAC layer 2. Network geometry and channel traces pre-generation 3. User traces pre-generation 4. Object-oriented programming (OOP)-based implementation 5. Optimizing the aggregate interference 6. Lite simulation 7. ROI gets split into a set of pixels of a fixed size 8. Parallel processing
WiSE [11]	1. Pre-generation of MIMO pre-coding matrices 2. Smart beam forming
X. Wang et al. [78]	None

channel traces, respectively while SiMoNe implements an efficient uplink abstraction based on [79]. On the other hand, 5G-air-simulator and Vienna 5G SL Simulator optimize run time through the use of object-oriented programming (OOP)-based architecture. To increase time efficiency in MIMO-based simulations, WiSE combines pre-generation of MIMO precoding matrices and smart beam sweeping link selection.

2.3.5 Evaluation Summary and Simulator Applicability

The evaluation presented earlier shows how close each simulators are to breaking this iron triangle. Table 2.6 summarizes the degree of realism, completeness, and computational efficiency of the current 5G system-level simulators, and thus show how close each simulators are to become DT ready. It should be emphasized that this rating is solely based on the open-source user manuals and literature. This rating excludes any updates that are not reported in user manual and literature at the time of writing. Similarly, the 5G components that are not specified in the open-source literature of simulators are denoted by (-) in the table. Finally, we acknowledge that the rating score is not intended to rank the simulator as each has distinct strengths, but rather to assist researchers in identifying the most relevant simulators for their respective works.

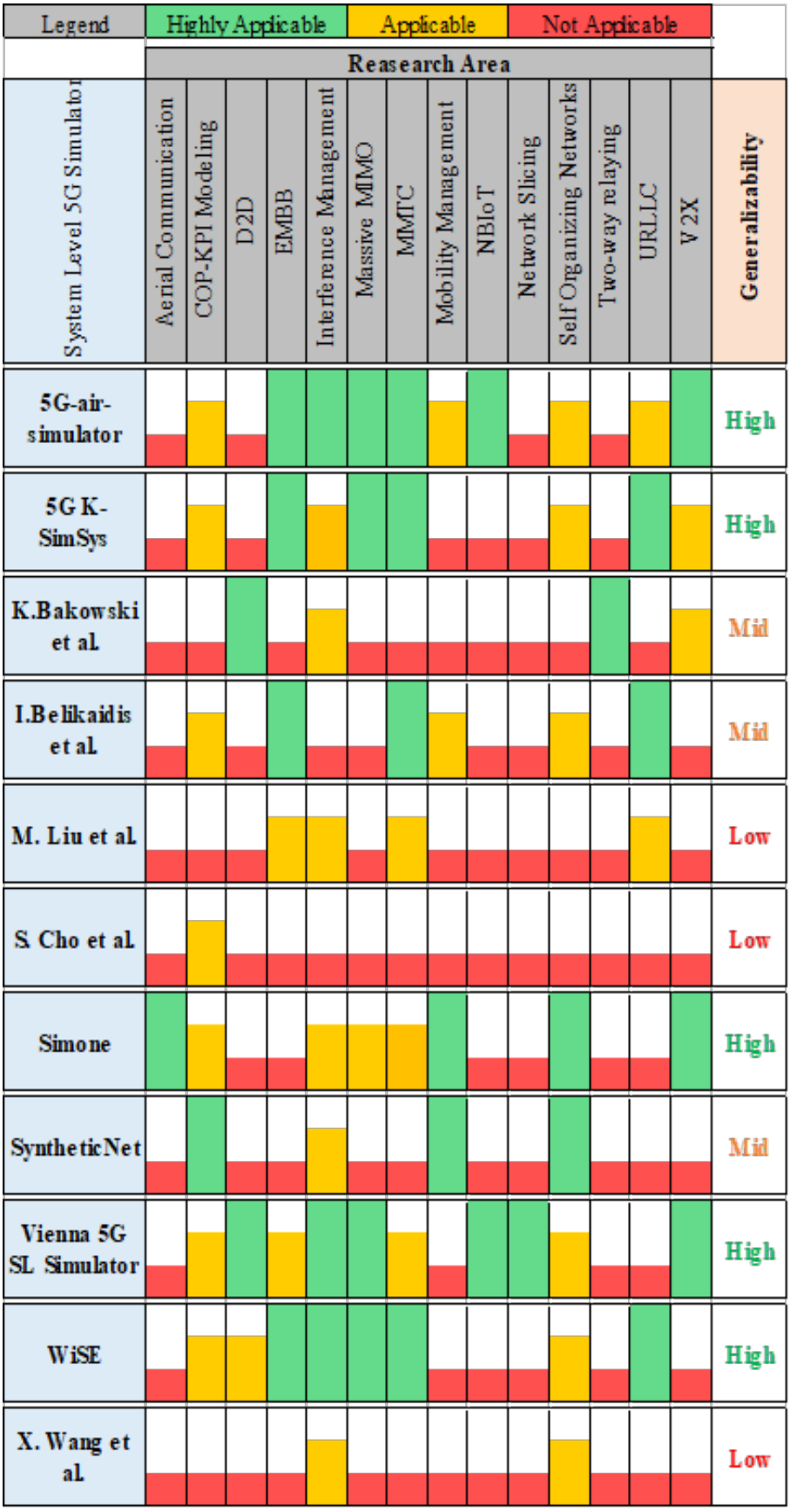
Table 2.6: Evaluation of the readiness of 5G system-level simulators to act as DT kernel based on insightful 5G-specific metrics defined in Table 2.2

5G-Specific Metrics \ 5G System-Level Simulators		5G-air-simulator	5G K-simsys	I. Belikaidis et al.	K.Bakowski et al.	M.Liu et al.	S. Cho et al.	SiMoNe	SyntheticNET	Vienna SL 5G Simulator	WiSE	X. Wang et al.
REALISM	Propagation	★★	-	★★	★★	★★	-	★★	★★	★★★	★★★	★★
	Channel Model and Calibration	★★★	★★★	★★	★★	-	-	★★★	★	★★	★★★	★★
	Handover Model	★	★	★	★	★	★	★★★	★★★	★	★	★
	User Mobility Model	★★	-	★	★	★	-	★★★	★★	★★	-	-
	Traffic Demand Model	★★★	★★	★★★	-	★★	-	★★★	★	-	★★	-
	Network Geometry	★	★★	-	★	-	★	★★	★★	★★★	★	★
COMPLETENESS	Implementation of 5G features mentioned in Table 2.7.	★★★	★★	★★	★	★	★	★	★★	★★	★★★	★
COMPUTATIONAL EFFICIENCY	Methods implemented to improve the computational complexity of running simulations.	★★	★	★	★	★	★★	★★	★★	★★★	★★★	★

The evaluation shown in Table 2.6 highlights that SiMoNe has the highest degree of realism owing to the realistic modeling of the channel, handover, user mobility, and traffic demand. Likewise, 5G-air-simulator, Vienna, and WiSE are not far behind in terms of realism. These simulators provide accurate representations of propagation and channel behavior. SyntheticNET has in-depth implementation of handover while *I. Belikaidis et al.* exhibits a high traffic demand model rating. Meanwhile, 5G-air-simulator and Vienna demonstrate a high level of completeness. Both simulators implemented five of the eight advanced 5G features listed in Table 2.2 indicating that these simulators are capable of closely mimicking several aspects of a real 5G network. K-SimSys, *I. Belikaidis et al.*, SyntheticNET, and Vienna receive two-star ratings for incorporating at least one advanced feature of 5G while the remaining simulators received a single star rating. Finally, Vienna and WiSE emerge as the most computationally efficient simulators, utilizing multiple techniques to minimize the computational complexity associated with running simulations.

Table 2.7 summarizes the applicability of each simulator in major current and emerging research areas of cellular networks. This applicability is based on the insights gained from the evaluation using the proposed metrics. This table is intended to provide assistance to the research community in selecting the appropriate simulator based on their research interests. We categorize the applicability of each simulator in three levels: (1) highly applicable, (2) applicable, and (3) not applicable, based on the implemented features and functionalities. For instance, simulators with a rich selection of traffic demand models, i.e., 5G-air-simulator and SiMoNe can be used to conduct experiments on different use cases such as eMBB, MMTC, and URLLC that requires a variety of traffic scenarios. Similarly, the 3GPP-based handover implementation offered by SiMoNe and SyntheticNet makes them an excellent

Table 2.7: Applicability of different 5G system-level simulators to different research topics.



choice for research on mobility management. Finally, unique features such as side-link support of Vienna, ground-to-air link support of SiMoNe, and novel design of *K.Bakowski et al.* make them highly applicable in D2D, aerial communication, and two-way relaying scenarios, respectively.

The evaluation metrics proposed in this study reveal a trade-off between different factors, which creates a unique challenge of extreme computational complexity. This challenge is expected to be amplified in the context of 6G technology.

2.4 Paving the Way Towards 6G: Future Requirements and Challenges of the Next Generation Simulators

Although no standard has yet been crafted for 6G, based on the current trend, it is not difficult to anticipate where 6G is heading and what it will look like. At this early point in time, several studies are available in the literature ranging from the anticipated architecture, potential applications and use cases, to the expected enabler technologies to support 6G summarized in Fig. 2.5 [80–84]. In terms of the architecture, one of the most evident distinctions of 6G from 5G is the utilization of diverse network types in the form of 3D networking. In addition, 6G is likely to sustain a much broader range of use cases and applications compared to 5G. Some of the most notable novel applications are flying cars, the internet of everything, multisensory extended reality (XR), and wireless brain computer interaction. To realize these use cases, the next generation networks will leverage novel sets of enablers such as ultra-massive MIMO, Thz band, quantum communication, blockchain, optical wireless technologies, and reflective surfaces.

In the wake of the anticipations, it is evident that 6G requires trial, evaluation, and validation procedures, that are more flexible and extensive than they have ever been. As a result, the simulators needed to support the timely development of the incoming 6G network deserve more attention. Particularly, modeling the above-

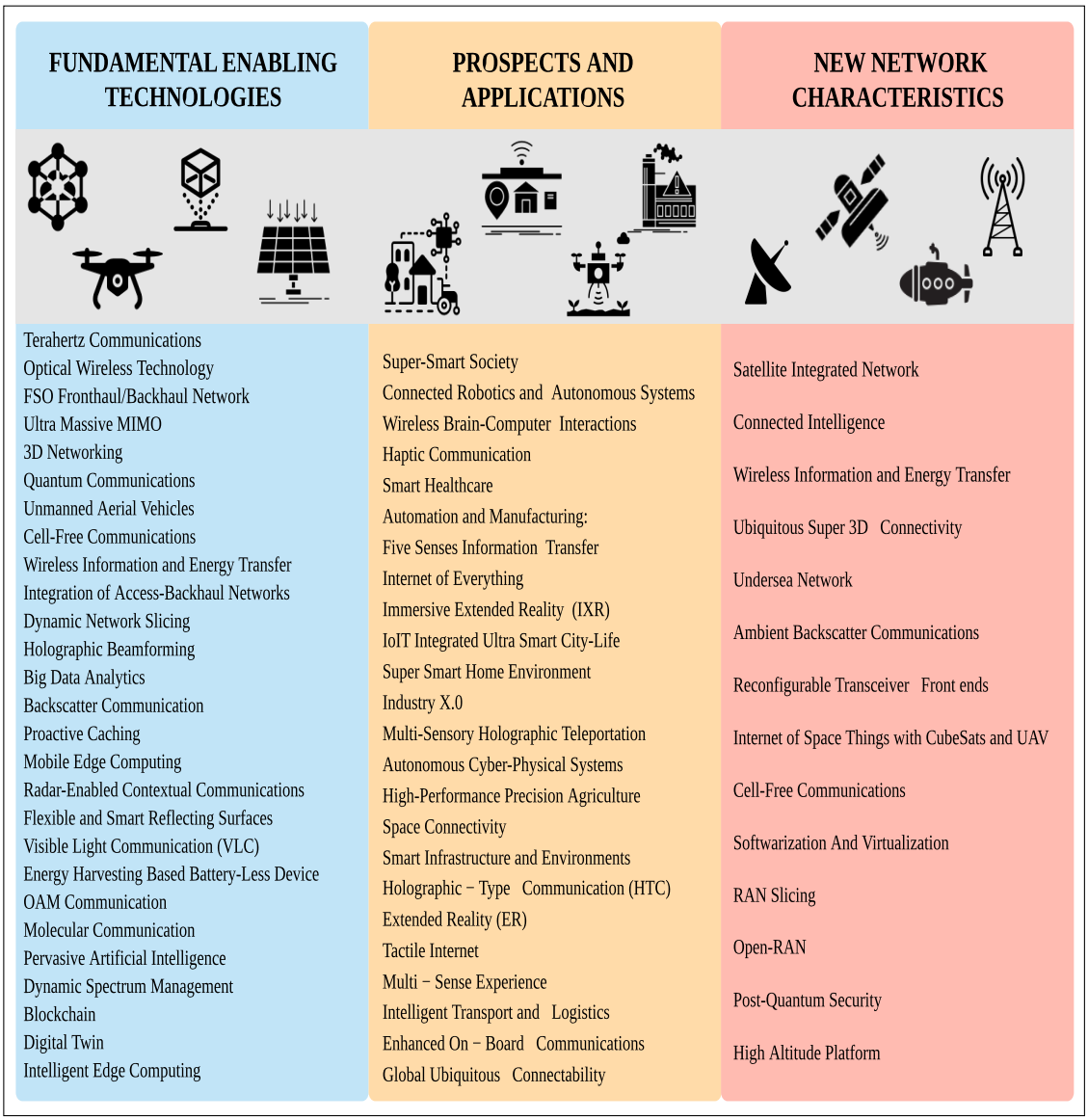


Fig. 2.5: Summary of enablers, prospect applications and novel architecture design of the incoming 6G network.

mentioned architecture, use cases, and enablers of 6G realistically and completely brings unprecedented impediments to the computational efficiency of next generation simulators.

2.4.1 Extreme Computational Complexity in 6G

The computational efficiency requirement for 6G simulators conflicts with the goal of creating simulators that are as realistic and complete as possible. All three of these are not only desirable features for the research and development of 6G systems but are also key necessities for generating synthetic data to train AI for enabling zero-touch operation and optimization in 6G networks. Fig. 2.6 shows the relationship between realism, completeness, and computational efficiency of system level mobile network simulators. To avoid unbearable computational costs, most of the existing simulators tend to employ more abstraction and oversimplification of the computationally intensive tasks and incorporate fewer features and functionalities, thereby compromising realism or completeness or both. Some well-known academic simulators, such as ns-3 [85] and Vienna [77], have managed to render a moderate amount of realism and completeness, but at the cost of high computational costs. On the other hand, commercial simulators such as Atoll [86], are usually utilized in network planning and thus, give particular attention to realistically modeling the propagation model. But to improve the computational efficiency, these commercial simulators usually eliminate the support for mobility or handover and usually implement only a handful of COPs and KPIs thereby offering little completeness. Despite the good intention, the simplification and downscaling of the nuances during simulator development to induce efficiency in the run time may lead to undesirable results.

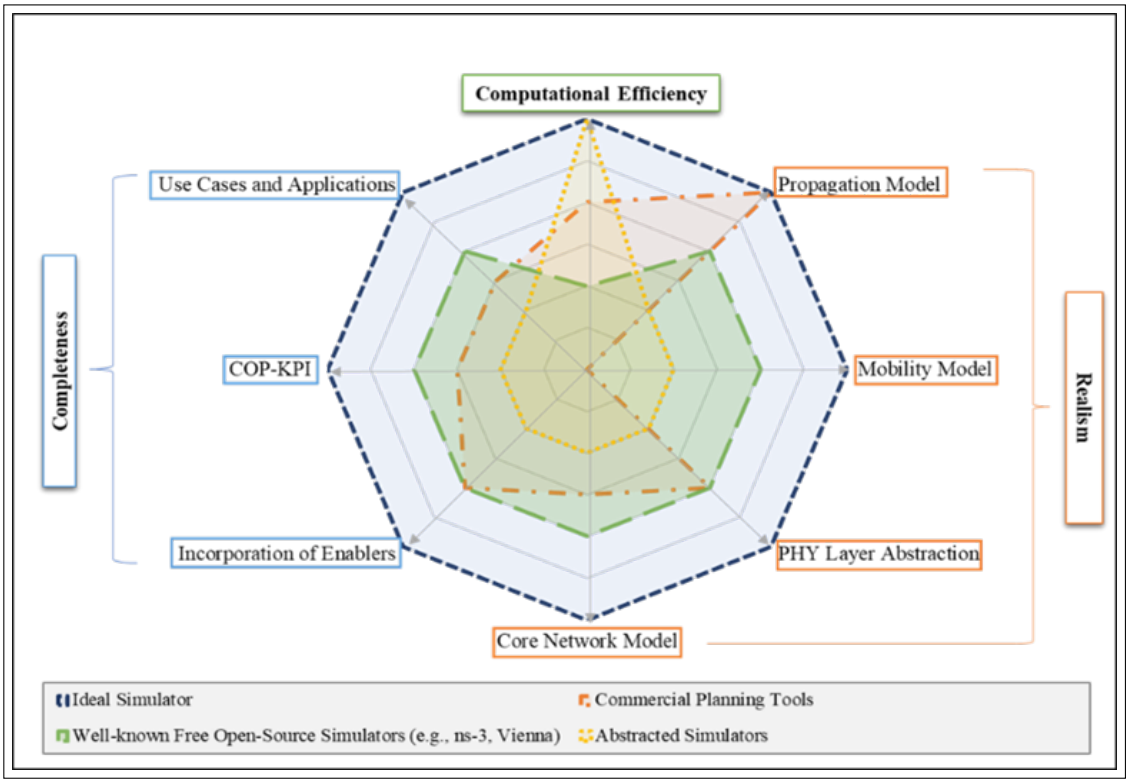


Fig. 2.6: Relationship between realism, completeness, and computational efficiency in mobile network simulators

2.4.2 Computational Demanding Aspects of Running Simulations in Future Networks

Table 2.8 shows the summary of some of the most computationally demanding tasks in simulating a cellular network. Among these components are the propagation model, MIMO precoding, mobility, network element size, compulsory use of AI, and reconfigurable intelligent surfaces. Moreover, this table also shows how these components are affecting the current 5G simulators and more importantly, their implications for the upcoming 6G technology.

Selected computationally challenging components for running simulations are explored in further detail below:

- Propagation Model: The increasing number of factors that must be considered

Table 2.8: Summary of some of the most computationally demanding aspects when running a network simulator, how it affects simulation run time in simulating 5G network and the future implications on the upcoming 6G networks.

Computationally Challenging Components	Details	5G network	6G network
Propagation Model	As operating frequency increases, more parameters in propagation modeling must be considered to improve accuracy (i.e., type of clutter, vegetation, buildings, people, and cars), making it computationally expensive.	mmWave	THz band
MIMO Beam Management	The increased computational complexity introduced by the increased number of antennas used to achieve MIMO is related to the increased number of precoding matrices required to generate during the simulation.	Massive MIMO (16x16 to 64x64 antenna arrays)	Ultra-massive MIMO (1024x1024 antenna arrays)
Mobility	The constant variation in the location and environment as the users move leads to continuous updating of the signal strength during simulation.	2D mobility	3D mobility
Number of network elements and connected devices	The size of network elements such as base stations, connected devices, configuration parameters, measured KPIs, and supported features directly affects the simulation run time.	BS Parameters: 2500+ User density: 10^6 devices/km^2	BS Parameters: 3000+ User density: 10^7 devices/km^2
AI-model training	Training AI-based model is computationally expensive depending on scale of the model (i.e., model depth), and size of the data.	Arbitrary	Compulsory
Reconfigurable Intelligent Surfaces	Calculation and delivery of all active beamforming weights and phase shifts to the associated nodes.	N/A	Among the enablers
Network Deployment	The computing requirements for the aforementioned components exacerbate with the introduction of non-traditional network deployment.	Terrestrial (2D Networking)	Terrestrial + Aerial (3D Networking)

in the propagation model to increase accuracy (i.e., clutter type, vegetation, buildings, etc.) make it computationally expensive. In mmWave used for 5G, factors such as building materials absorption, vegetation, vehicles and even the effect of humans as blockage should be considered which further increases the complexity of the process. The incoming 6G will require an unprecedented amount of sophistication when it comes to propagation modeling. This is due to the anticipated utilization of the THz band which, compared to the GHz band used in 5G, is more demanding in terms of line of sight (LoS)

requirements and is highly susceptible even to the slightest obstruction (i.e., a sheet of paper) [82]. To realistically model these peculiarities of the THz band requires considering more factors and thus, would require immense computing resources.

- MIMO Precoding/Beam Management: The multiplicity of the utilized antennas to realize MIMO brings additional complexity due to an increase in the precoding matrices needed to be generated during the simulation. The available antenna configurations for 5G massive MIMO range from 16x16 to 64x64. With this much antenna, precoding matrices generation consumes more time when running simulations. The concept of ultra-massive MIMO is proposed for 6G, wherein a plasmonic nano antenna array of size 1024x1024 is envisioned [80]. With this huge antenna configuration, the generation of precoding matrices will be daunting for the simulators.

- Mobility: The constant variation in the location and environment as the users move leads to continuous updating of the signal strength during simulation. In addition, mobile users need to perform handover, which is also computationally demanding. 5G is designed to support mobile users with a maximum speed of up to 500 kph. Calculation of the changes in the signal strength, especially for high-speed users, is an expensive task. Moreover, due to speed as well as the dense base station deployment, the number of handovers increases. The maximum speed at which 6G will be molded is 1000 kph, making the calculation of the changes in the received signal more daunting.

- Huge Number of Network Elements and Connected Devices: The number of network elements such as base stations, users, types of services, connected devices, configuration parameters, measured KPIs, and supported features directly affect the simulation run time. A recent article [8] highlights this dramatic increase in the simulation execution time with an increase in the

network size. Based on this article, the simulation time showed a non-linear surge in time complexity, increasing from less than 100 seconds for 1000 nodes to more than 2000 seconds for a 3000-node scenario. Meanwhile, it is estimated that the connectivity density in 6G will reach far beyond 106 km2, which is the density threshold for designing 5G [87]. With dense BS deployment supporting diverse types of users, use cases, and billions of connected devices, computationally inefficient simulators will fail to mimic the vastness of the 6G ecosystem.

- Network Deployment: Meanwhile, one of the major game-changers that set 6G apart from its predecessors is the introduction of 3D networking. The conglomeration of aerial and terrestrial base stations is expected to shift the way we analyze and simulate cellular networks. 3D networking is anticipated to bring a whole new level of complexity in simulating the network and making the aforementioned challenges even more daunting. For instance, the wide adaptation of drones will give birth to 3D handover. This, in turn, results in more sophisticated user mobility and handover model implementations.

2.4.3 Proposed Architecture to Address the Computational Complexity of Next Generation Network Simulators

Based on the above discussions, the usability and performance of a simulator for 6G are severely undermined if it fails to immediately test the use cases and generate results rapidly. Therefore, the new design, protocol, or algorithm evaluation, as well as the data generation through such a simulator can become time-consuming if not accompanied by computationally efficient methods. Currently, several methods to speed up the simulation process for 5G networks are being leveraged. For example, several simulators, such as Atoll, OMNET++ [88], MATLAB-Simulink [89], and SiMoNe [10], exploit parallel processing to improve the run-time of simula-

tions. Similarly, other simulators attempted to explore simulation speedup options to make up for high computational complexity either due to a large number of network elements, realistic modeling, or completeness. For instance, Vienna 5G has pre-generated channel traces while ns-3 tries to perform link-to-link computation in parallel. Meanwhile, WiSE incorporates MIMO precoding matrices pre-generation and smart beam sweeping link selection, in addition to parallel processing, to improve time efficiency [11].

Although some efforts are made to improve the computational efficiency of simulators, our extensive analysis of existing simulators shows that the bulk of the computational load is caused by their object-oriented architecture, where extensive iterative functions (e.g., for-loops) are used in each transmission time interval (TTI) to calculate KPIs. Taking this into consideration, there is a call for a major shift in designing a simulator that minimizes the use of iterative functions. In addition to the above-mentioned approaches, we highlight some key techniques for addressing computational efficiency while maintaining a high level of realism and completeness below.

- Pre-generation and Pre-loading of COP-dependent KPIs: The performance of cellular networks, measured in terms of Key Performance Indicators (KPIs) may vary depending on configuration and optimization parameters (COPs) settings. COPs are the backbone of any cellular network system. These tunable COPs, depending on the set values, affect how the network performs. For instance, COPs such as tilt, azimuth, tower heights and azimuth determine several KPIs like coverage and reference signal quality that do not change with time-variant factors such as user mobility or channel variations but only with respective COP values. Therefore, these types of KPIs can be pre-calculated even before the start of the simulation. Meanwhile, other KPIs vary not only with COPs but also with time variant factors. Examples of KPIs include

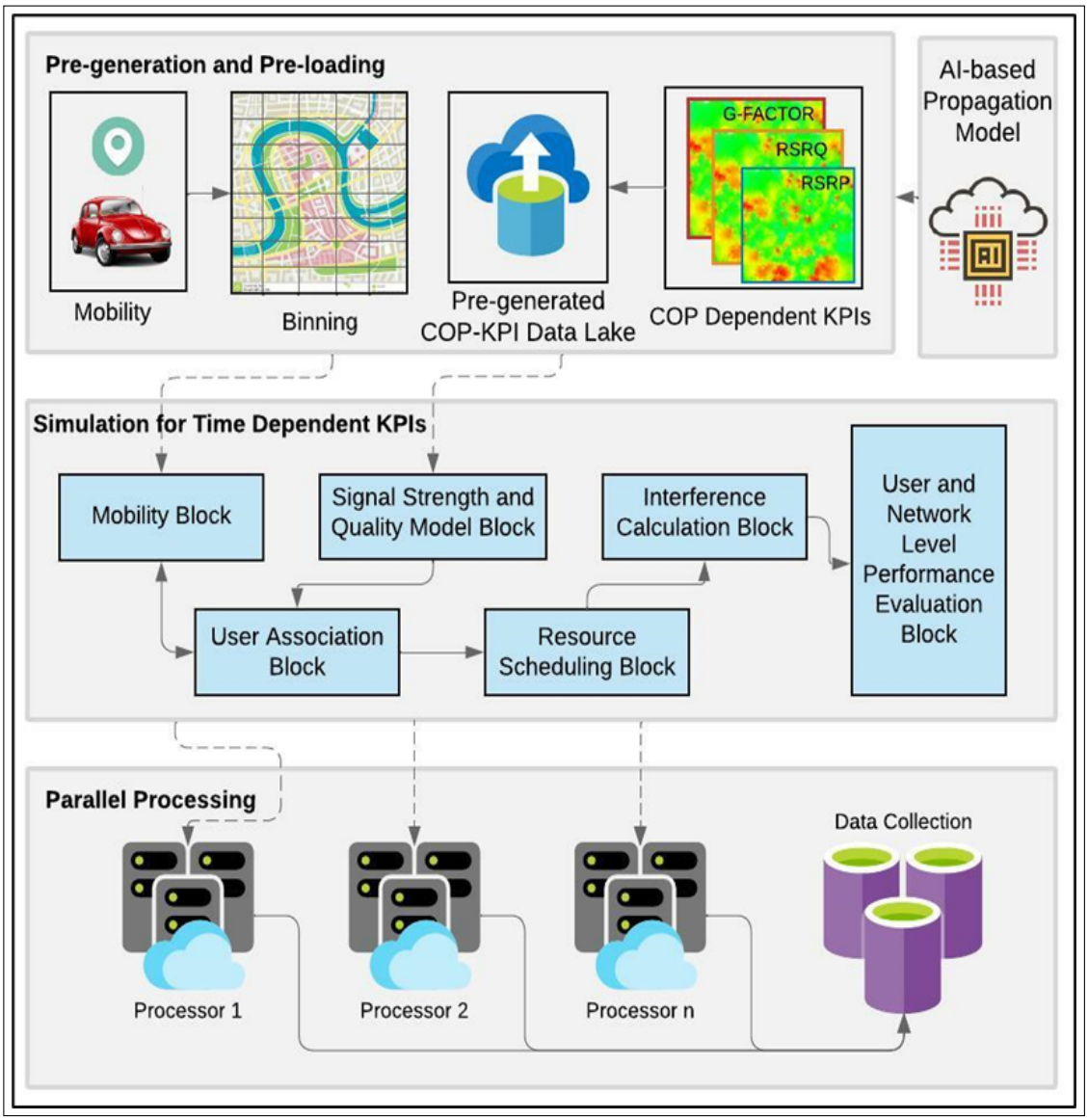


Fig. 2.7: Proposed Python-based simulator architecture to enhance the computational efficiency of future simulators.

throughput, handover success rate and quality of service and user experience. Although these KPIs are also affected by COPs, their actual values cannot be predetermined unless the simulation is started, and users start to move around the network and request resources. Having this knowledge, instead of calculating all KPIs in each TTI, the simulator design can divide KPI modeling into two categories: time-dependent and COP-dependent. All COP dependent KPIs such as reference signal received power (RSRP), reference signal received quality (RSRQ), geometric signal-to-noise-plus-interference ratio (SINR) also known as G-Factor, and slow shadowing can be pre-calculated only once at the beginning of the simulation thus incurring only a small computation cost despite using extremely realistic models. These pre-generated KPIs can then be pre-loaded at the start of the simulation process. This leaves only time-dependent KPIs such as instantaneous user SINR, physical resource block (PRB) usage, handover evaluation metrics, and throughput to be calculated in each TTI.

Nonetheless, generating such voluminous amounts of pre-generated COP-KPI data would need huge storage space, particularly if flexibility in the simulated scenario, e.g., in user mobility, is desired. To overcome this issue, cloud storage may be a viable alternative. Instead of keeping the pre-generated data on a local system, they can be saved in a data lake on the cloud. This data lake will include a rich collection of network scenarios and will be expanded further as new scenarios are executed. When needed, the pre-generated network data can be readily retrieved from the data lake. Given that storage is usually cheaper than processing power and is not as constrained resource in simulators as the simulation run time, this solution can offer the desired degree of trade-off between the simulator's use case flexibility, storage capacity and run time.

- Modeling Mobility Innovatively Through Binning: In current simulators, es-

pecially those that are developed to provide a very high level of coverage prediction fidelity, such as Atoll, mobility modeling is avoided as it is computationally very expensive, i.e., the next location of each UE must be calculated in each TTI and all the KPIs must be recalculated with respect to the new location of the UE. However, being the raison d'etre of mobile cellular networks, mobility is an essential feature of mobile networks and must be considered to achieve holistic performance evaluation and optimal design that consider mobility related KPIs and not only static coverage predictions. Particularly optimal mobility management's contribution to system performance will increase drastically in 6G, where ultra-high user speeds of up to 1000 km/h are anticipated to be supported. To model user mobility while eliminating the associated computational cost, the binning approach can be leveraged. The key idea here is to divide the network area into bins (cubes in the case of a 3D network) and to model COP-dependent KPIs with respect to spatial bins in the network instead of the UE locations. Thus, pre-calculated COP-dependent KPIs, as explained above, can be used in each TTI for the bins with which the user is associated. This approach eliminates the need for recalculation of all KPIs in each TTI, even with a large number of mobile users. Secondly, the binning approach will require recalculation of UE-specific KPIs only when the UE changes bin location and not at every TTI. Finally, the bin size can be utilized to explore the tradeoff between the flexibility of the simulator in terms of accuracy and the volume of the pre-generated data. For example, a larger bin size can reduce the size of pre-generated data at the expense of KPI accuracy, and vice versa.

- Leverage AI to Model Computationally Intensive Tasks: Pathloss calculation is another computationally expensive task if modeled realistically, e.g., using raytracing. For instance, Atoll, by utilizing the ray-tracing model, yields far superior fidelity in coverage prediction than Vienna and ns-3, which use

simple empirical and hence unrealistic propagation models. However, to offset the huge computational cost of raytracing's, Atoll omits all dynamics in the simulation, such as dynamic PRB allocation and scheduling, as well as user mobility and detailed handover procedures and signaling. Therefore, Atoll, while being very powerful for cell planning, cannot be used for research and development of realistic mobile networks or optimization of any of the dozens of mobility related KPIs and COPs. In other words, the Atoll architecture, like many other simulators, trades realism in certain aspects for incompleteness to keep the computational cost low. To address this tradeoff in a more optimal fashion, the potential of AI is leveraged to model computationally demanding tasks such as radio propagation. This idea is recently demonstrated in a study [13] that shows it is possible to leverage machine learning to achieve comparable accuracy to raytracing in propagation modeling. Results from this recent study show that Light Gradient Boosting Machine (LightGBM)-based propagation model outperforms all empirical models (e.g., used in Vienna and ns-3) in terms of accuracy while being 12x faster than raytracing used in Atoll. Similarly, authors in [90] propose a practical and accurate channel estimation for cell-free mmWave Massive MIMO framework based on the fast and flexible denoising convolutional neural network.

- Enabling Parallelization and Distributed Processing: Although several existing simulators already support parallel and distributed processing, operations of the majority of the existing simulators remain for-loops-based that do not allow parallelization. To enable parallelization, the conventional for-loop approach should be avoided by replacing them with paralleled matrix manipulations for most calculations. For example, interference calculation per PRB in each TTI can be modeled as a parallel-able matrix operation. In addition, the pre-loading will enable the parallel processing of different COPs for COP-KPI combinatorial exploration and hence, reduce the time complexity of COP-KPI

data generation with higher computational resources.

- Build on a computationally efficient platform (i.e., Python): Most of the existing simulators are built in either C++ or MATLAB. For example, ns-3 is a C++ based simulator, but it requires a highly C++ specific skill set. The need for an extremely experienced and skilled workforce hinders the use of ns-3 to some extent despite its ability to utilize the high-performance computing power of C++. On the other hand, MATLAB-based simulators are relatively easier to learn, but open-source MATLAB lacks the high-performance computing power to fully utilize the available resources. This limitation makes the MATLAB-based simulators such as Vienna, Simulink, and C-RAN, relatively slow. However, Python possesses both the advantages of C++ and MATLAB; it is open source and utilizes high-performance computing like C++ and it is easier to learn like MATLAB. In addition, Python has a plethora of AI and DL libraries, which are far richer than those of C++ and MATLAB. This capability makes Python a better choice for the development of a system level 6G simulator, as it will enable easier integration and testing of AI solutions on emerging networks.

To accelerate the simulation time regardless of the size of the network, with high degree of realism and completeness, and to generate a large amount of dataset in a short amount of time, the approach towards developing future simulators demands an overhaul. In this regard, next generation simulators will require leveraging innovative simulator architecture, parallel, and distributed processing capabilities, as well as time and computationally efficient modeling of time complex functions like mobility and propagation. Fig. 2.7 illustrates the proposed architecture and techniques to improve the computational efficiency of future simulators. In summary, this framework aims to investigate and develop novel techniques to expedite simulation time, which include: 1) a novel architecture leveraging the fast and efficient

matrix implementation instead of for loop-based structure; 2) reducing the number of computational and time-hungry calculations of user mobility traces, RSRP, RSRQ, and SINR in each TTI; 3) utilizing ML-based pathloss models with significantly less time complexity than ray-tracing based pathloss model; 4) exploring innovative ways to model user mobility in a large cellular network; 5) fully utilizing the powerful parallel processing capabilities on one client without compromising the quality of the data; 6) proposing a decentralized simulator architecture to integrate multi-client data generation.

2.4.4 Case Study: Efficient Generation of Synthetic Network Data for Training AI Models

To quantify the advantage of the proposed simulator architecture over the state-of-the-art architectures, this section examines the data generation time in three types of simulators: (1) a legacy simulator, (2) a simulator with parallel processing, and (3) our proposed architecture-based simulator, i.e., a simulator that leverages binning and pre-generation, paired with parallel processing to reduce run time/online computational complexity. Specifically, the use case demonstrates the increase in the simulation runtime as the number of nodes (i.e., connected devices) increases and quantifies the potential of the proposed architecture to address this issue. As 6G is anticipated to support a greater number of connected devices, the case study analyzes the capabilities of various simulation techniques/architectures to accommodate a high user density in terms of simulation completion time. In addition, the use case highlights the effectiveness of the proposed solution in generating data for more effective machine learning model training. This is accomplished by evaluating the performance variation of AI solutions trained on data generated using different simulation strategies.

Simulation Setup

We exploit a 3GPP-compliant, state-of-the-art system-level simulator named SyntheticNet [1] [63] to validate our proposed simulator architecture. SyntheticNet is a modular, flexible, and versatile simulator supporting advanced features like adaptive numerology, handover, and futuristic database-aided edge computing, to name a few. In this article, we have created three versions of SyntheticNet. The first version is the legacy simulator, which has none of the aforementioned innovative approaches implemented in it to improve the computational efficiency. In the second version, we equip off-the-shelf SyntheticNet with the ability to perform parallel processing, similar to the current state-of-the-art approach of other simulators such as OMNET++, Simone, MATLAB-Simulink, and Atoll. In this version of the simulator, we allocate 20 cores capable of running the simulation in parallel. Finally, in the third simulator version, we implement the proposed innovative approaches to reduce computational complexity, such as pre-generation and pre-loading of COP-dependent KPIs and mobility modeling through binning. In addition, this version is also capable of performing parallel simulations.

For the first part of the use case, we run the three versions of the simulator using similar settings as shown in Table 2.9. To see the impact of varying the number of network elements in the simulation runtime, we vary the number of users from 100 to 2000. Each simulator version is run to generate data equivalent to 15 seconds of real network data. The generated data is composed of several combinations of COPs: A3-Offset, A3-Time to trigger (TTT), A2-Threshold, A2-Time to trigger (TTT) with the corresponding KPI (Throughput) for each of the combinations. In total, each simulator version generated data of around 3,575 combinations of COP-

[1]SyntheticNET is a system-level simulator developed by the AI4Networks Research Center. The present work builds upon previous efforts to enhance SyntheticNET's computational efficiency, resulting in a fully optimized and functional simulator. The innovative solutions presented here are the result of a collaborative effort involving numerous experts from the AI4Networks Research Center.

Table 2.9: Network simulation settings.

Parameters	Values
Number of Base Stations	100
Number of Users	100, 200, 500, 1000, 2000
Number of COP-KPI combinations	3,575
COPs	A3-Offset, A3-TTT, A2-Threshold, A2-TTT
KPI	Throughput
A3-Offset	[0, 1, 2, 4, 5, 6, 7, 8, 9, 10] dB
A3-TTT	[64, 128, 256, 512, 640] ms
A2-Threshold	[-95,-97,-99,-101,-103,-105,-107, -109,-111,-113,-115,-117,-119] dB
A2-TTT	[32, 64, 128, 256, 512] ms
Simulation Time (Data Generation)	15 s

KPI. Meanwhile, for the second part of the use case, each of the simulator versions is run for 1 hour with 100 users. After this period, simulations are stopped, and the data generated are gathered to train AI models.

Run-time Comparison and Analysis

The first set of results is the comparison of the total run time of the three simulator versions shown in Fig. 2.8. As expected, for all versions of the simulator, the run time increases as the number of users increases. More importantly, results reveal that as user density increases, traditional simulators require exponentially higher time to complete the simulation compared to the simulator with the proposed approach. From almost 100 hours for 100 users, the run time grows to more than 2166 hours. This means the legacy simulator architecture cannot be scaled to simulate 6G networks. Meanwhile, a simulator with parallel processing demonstrates it can better deal with the higher user density compared to the legacy simulator. Since the simulations are run in parallel using 20 cores, the simulation time is reduced by 20x the original. Simulator with parallel processing takes around 5 hours for 100

users and approximately 109 hours for 2000 users to finish the simulation. Although parallel processing showcased the ability to cut the runtime, the simulator with the proposed innovations surpasses its performance. With the proposed approach, a simulation with 100 UEs takes only one hour to complete, whereas a simulation with a maximum of 2,000 UEs requires around 21.7 hours to generate the data. In summary, the performance of the proposed simulator architecture is 5x faster than the state-of-the-art parallel processing approach and 100x faster than most legacy simulators that do not allow parallel processing. These results demonstrate the potential of the proposed architecture to simulate networks with high user density scenarios that are the hallmark of 6G. While this is a considerable improvement over the legacy simulator, the fact that the simulation still requires a large amount of time (1 hour for 15000 TTIs) leaves a lot of innovation margin before real-time network emulations can be realized. In addition, for a simulator to have the potential to act as a digital twin, it should have a comparable run time with the real network (i.e., 1ms for 1 TTI).

AI Model Training Analysis

One of the potential applications of AI in 6G is the prediction of network behavior. AI models have the capability to model and map out functions that cannot be directly or mathematically interpreted in the data. With available data, these models give insights into the traffic patterns and network behavior. However, to effectively train AI models, a representative amount of data is needed. In this use case, we analyze the effect of data sparsity caused by the inefficient data generation capabilities of traditional simulators. We build models that can map out the relations between mobility-related parameters, i.e., A3-Offset, A3-TTT, A2-Threshold, A2-TTT against certain KPI, i.e., throughput. The performance of the model is measured by the root mean square error (RMSE) metric. Lower values of RMSE

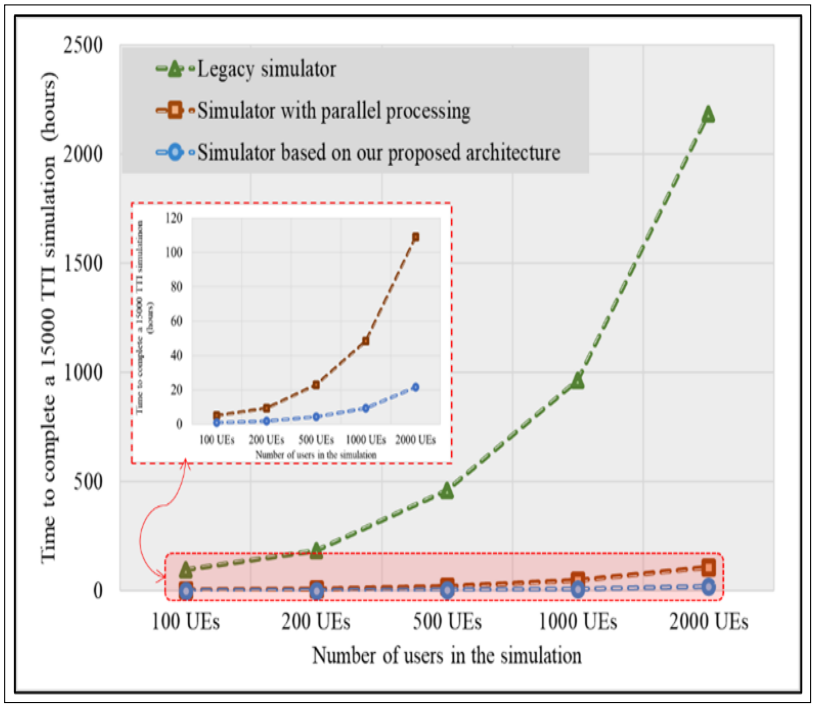


Fig. 2.8: Run-time comparison between simulators with and without the proposed method to improve computational efficiency.

correspond to a well-trained model, while large values indicate insufficient training. We train and evaluate several AI models, namely Linear Regression, Polynomial Regression, Support Vector Regression (SVR), Decision Tree, Random Forest, and XGBoost using the data generated by the three simulator versions.

The second set of results shows the effect of the simulators' capacity to generate the required amount of data needed to train machine learning models. As legacy simulators are inherently slow, they generate the least data points in 1 hour runtime (50 combinations of COP-KPI data), followed by the simulator with only parallel processing (700 data points). Lastly, due to its computational efficiency, the proposed simulator generates more than 3500 data points in 1 hour. We use the datasets generated by the three simulation techniques separately to train AI models for throughput prediction and compare the performance in Fig. 2.9. It can be observed that the models trained on the data generated by the proposed simulator architecture have the lowest RMSE. The average RMSE of the models trained with

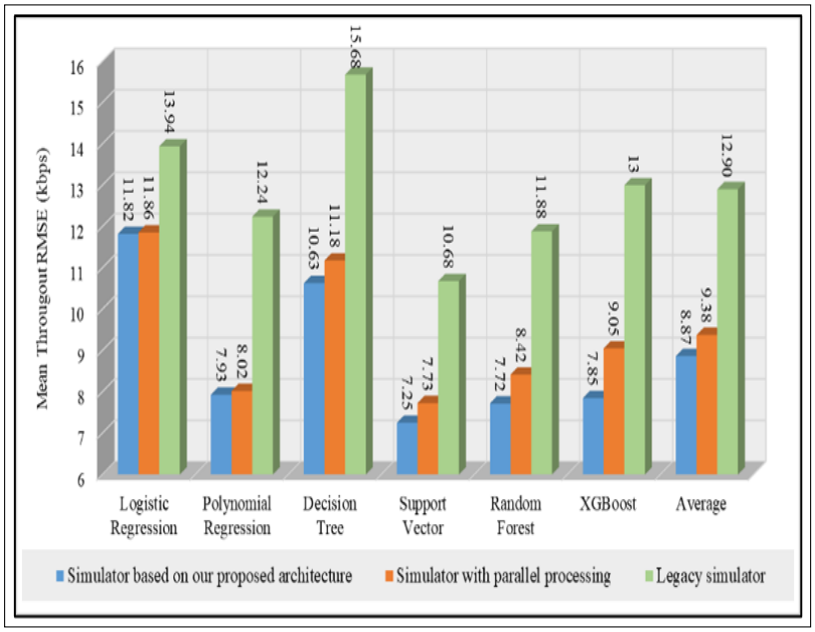


Fig. 2.9: Comparison of AI models trained with different amounts of data generated from the legacy simulator, parallel processing-capable simulator, and simulator based on our proposed architecture.

a large amount of data using the proposed architecture is 8.87 kbps, better than the 9.38 kbps and 12.90 kbps average RMSE of legacy and only parallel processing-capable simulators, respectively. This implies that the proposed architecture-based simulator is more capable of enabling the training of AI models for existing (i.e., 5G) and emerging complex cellular networks such as 6G. This further highlights the utility of the proposed architecture for various R&D use cases for 6G, particularly for aiding data driven modeling and AI-based network optimization.

2.5 Conclusion

Simulators play a crucial role in the development of cellular network technology, especially with the advent of digital twin in cellular networks, which has increased their utility. In this chapter, we conduct a comprehensive literature review of over 17 existing system-level simulators for 5G and beyond (5G&B) networks and present a comparison of their features and capabilities to raise awareness among researchers

and practitioners. Furthermore, we highlight the limitations of current metrics used for evaluating simulators and propose novel 5G-specific metrics for evaluating system-level simulators, focusing on realism, completeness, and computational efficiency. These metrics enable us to assess the readiness of simulators for creating Digital Twins (DT) and provide insights for the research community to select the most appropriate simulator for their unique use cases. We provide insights into the enablers, prospective applications, and novel architecture design of future 6G networks, highlighting the forthcoming challenges in the development of next-generation simulators particularly the extreme computational complexity. TO address this challenge, we propose a novel Python-based simulator architecture that aims to transform the computational efficiency of simulators for 6G networks. We evaluate the efficacy of the proposed architecture through a use case and demonstrate its superiority over state-of-the-art approaches, not only in terms of higher computational efficiency but also as a better enabler for AI-based model training using synthetic data.

CHAPTER 3

Design and Development of Testbed for Realistic Mobility Evaluation

3.1 Introduction

While the computationally efficient simulator presented in the previous chapter can address the challenges of huge data generation needed to train AI-models to develop mobility management schemes, it may still lack in complete system realization. To address this limitation, testbeds are a promising solutions to enable real world evaluation of solutions to enhance mobility. However, testbed deployment can be a complex and time-consuming process involving many cascaded design procedures, feasibility studies, and making multi-faceted trade-offs through datasheet-driven and budget-constrained equipment selection decisions. It requires proper planning and insights to guide the design and deployment. In this chapter, we discuss the design considerations and challenges one might face while deploying an outdoor 5G&B system-level testbed. In particular, we present our hands-on experience of designing, planning, and deploying a 5G&B testbed called TurboRAN: Testbed for Ultra-Dense-Multi-Band Control and Data Plane Split Radio Access Networks of the Future [1]. TurboRAN is built using SDR-based 5G base stations also known as gNodeB (gNB). Deploying such a testbed for indoor experimentation or a limited outdoor setup is relatively straightforward. However, configuring it in an outdoor setup to cover a wide area with continued over-the-air coverage to enable system-level solution testing is quite challenging [91, 92].

[1]The creation of TurboRAN was made possible through the collaboration of multiple teams and experts from diverse domains, both within and outside of OU-Tulsa.

3.1.1 Contributions and Organization

- First, we present a summary of the currently available 5G&B testbeds around the world. This summary provides key information about several testbeds, including the deployment and access type. Moreover, we provide a discussion and highlight their key features, capabilities, and the different experiments they support. Although non-exhaustive, the presentation of several testbeds creates awareness for the readers regarding the current and emerging platforms that can be leveraged for the specific field of research.

- We then discuss the planning and design considerations when deploying a system-level testbed based on our hands-on experience of working on the TurboRAN testbed installation. This detailed discussion is focused on the aspects of testbed deployment, including link-budget, cell planning, selection of base stations, antennas, feeders, amplifiers, enclosures, and cooling systems among others.

- Finally, we show some preliminary results from our deployed TurboRAN testbed. This use case highlights the capabilities of the TurboRAN to support experiments on system-level problems such as mobility management. In the absence of over-the-air system-level multi-cell testbeds, this type of experiment would not be possible. Such experiments cannot be performed in a real network without causing service degradation. Results from the use case also shed light on the detrimental impact of sub-optimal handover parameter configuration and thus highlight the potential of testbeds to assist in the development of system-level optimization solutions otherwise difficult to investigate problems such as mobility management.

The rest of this chapter is organized as follows: In Section 3.2 we present some of the current and emerging 5G testbeds. In Section 3.3 we present a high-level

overview of the TurboRAN testbed, followed by the discussion of the deployment design and planning considerations. Meanwhile, Section 3.4 presents a case study regarding mobility-based inter-frequency handover, and Section 3.5 concludes the chapter.

3.2 Testbeds for 5G and Beyond Cellular Networks

The Third Generation Partnership Project (3GPP) has already completed the initial phase standardization of 5G and Phase I of 5G is already being deployed. The standard will keep evolving and the updates will be reflected in the new release. To support this evolution, it is important to investigate new features proposed by the research community and industrial partners. To avoid the risk of performance degradation when performing tests on real networks, testbeds are used as a practical solution. Testbeds are designed to function in the same manner as a real network but on a smaller scale. This means that, at least in theory, testbeds are capable of executing almost all the necessary functions of a real network such as downlink, uplink, handovers, resource allocation, and scheduling to name a few. Testbeds offer the freedom of experimenting with different scenarios and can also provide open access to the data generated from these experiments without privacy concerns. Given the importance of testbeds, it is imperative to create awareness regarding the existing testbeds the research community can utilize. In this section, we discuss different types of 5G testbeds that are available for different purposes.

There exist several ways to classify testbeds including the size of deployment (e.g., indoor, outdoor small-scale, city-scale) and type of access (e.g., open or proprietary). Another way of classification is by using the different layers of the OSI model. Based on these classifications, the research community divides the testing of wireless systems into two main levels: link-level and system-level. The link-level testing is related to the physical layer (PHY) as well as some functionalities of the

medium access control layer (MAC) and involves channel estimation, channel coding, rate matching, multicarrier modulation, and feedback techniques, to name a few [93, 94]. In contrast, system-level testing takes care of upper layers and focuses on system-level multi-cell operations such as handover, interference management, admission control, link adaptation, power control, and resource scheduling and allocation among others [67].

Several wireless testbeds have recently been established around the world to aid research on future cellular networks. Most prominent among these include the 5GIC [95], Ericsson's 5G Testbed [96], 5TONIC [97], FOKUS [98], NITOS [99] and SK Telecom 5G Playground [100]. Table 3.1 summarizes the present and emerging 5G and beyond network testbeds around the world. Among these, testbeds that are truly focused on system-level research aspects of the next-generation mobile networks primarily exist in Europe and Asia (i.e., [95–104]). The 5GIC testbed at the University of Surrey in the United Kingdom is one of the deployed testbeds closest to a real network. It is comprised of a full-fledged Long-Term Evolution (LTE)-based cellular network spread over four square kilometers. However, the utility of this testbed is bounded by two factors: 1) Cost—platinum level access is required to conduct the proposed research which is advertised at a membership fee of £600K per year or over $1M per year [105]; and 2) A commercial vendor administers and manages the testbed equipment with proprietary application programming interfaces (APIs) limiting its flexibility and programmability. As a result, academic researchers are unable to experiment on network characteristics directly and readily. Instead, they must rely on the equipment manufacturer to configure experimental parameters.

In the United States, there is a small number of open-access 5G testbeds. These include Cognitive Radio Network Test (CORNET) at Virginia Tech [106], AERPAW deployed in North Carolina State University [107], and 2 testbeds namely POW-

DER and PhantomNet located at the University of Utah [108, 109]. CORNET, as the name suggests, is mainly aimed to conduct experiments related to opportunistic spectrum access in the context of cognitive radio networks. The nodes in CORNET can be linked through a sophisticated channel emulator RFnest. Notably, the CORNET community wireless testbeds primarily feature programmable PHY and MAC layers and do not have the end-to-end, multi-cell programmable cellular network capabilities needed to conduct research on the themes identified in Table 3.2.

PhantomNet is a testbed dedicated to support 3GPP-compliant cellular network research [108]. This testbed offers emulation capabilities of eNodeB and user equipment (UE). It also has the ability to support software defined network (SDN)-related studies, making it a good platform for LTE core network exploration. It also includes Access Points and UEs based on Universal Software Radio Peripheral (USRP) to facilitate PHY layer research. However, PhantomNet is not based on actual cellular deployment. Instead of using a true air interface, PhantomNet connects RF devices (UEs and eNodeBs) using a custom-built RF attenuation matrix. On the other hand, POWDER [109], a city-wide testbed based at the University of Utah, is designed to do large-scale evaluations on SDNs and massive MIMO. Finally, AERPAW [107] is a one-of-a-kind testbed designed to investigate 5G network applications related to unmanned aerial vehicle (UAV).

3.3 TurboRAN Overview, Design, and Planning Considerations

In this section, we present the TurboRAN testbed by providing an overview of its deployment design, key components, and architecture. We present a summary of the steps considered for equipment related design choices and considerations as shown in Fig. 3.1. We then discuss in detail the design strategy, planning considerations, as well as the challenges we encountered during the deployment phase of TurboRAN.

Table 3.1: 5G network testbeds around the world.

5G Network Testbed	Location	Scale of Deployment	Access	Key Features and Supported Experiments
5GIC [95]	ICS, University of Surrey, Guildford, UK	University-wide	Close	IoT, broadband mobile radio, mmWave, satellite backhauling
5GUK Test Network [101]	University of Bristol, UK	City-scale	Not Specified	mmWave, network function virtualization (NFV), massive MIMO, massive IoT, network slicing
5TONIC [97]	Madrid, Spain	Indoor	Open	Network function virtualization (NFV)
Aalto 5G network [102]	Otaniemi, Espoo, Finland	University-wide	Open	NB-IoT, network slicing, mobile and edge computing, VR/AR, gaming, industrial Internet
AERPAW [107]	North Carolina State University, USA	City-scale	Open	UAV, mobility
CORNET [106]	Virginia Tech University, USA	Indoor	Open	Cognitive radio techniques, software-defined radio (SDR), dynamic spectrum access technologies
COSMOS [110]	West Harlem (New York City), USA	City-scale	Open	mmWave, distributed edge cloud, backhaul research
Ericsson 5G [96]	Ericsson, Stockholm, Sweden	City-scale	Close	5G-LTE dual connectivity, distributed MIMO, mmWave, massive MIMO
FOKUS [98]	Fraunhofer FOKUS and TU Berlin campus, Germany	University-wide	Open	Network slicing, URLLC, IoT
LuMaMi [103]	Lund University, Sweden	Mobile Base Station	Open	Massive MIMO
NITOS [99]	University of Thessaly (UTH), Volos, Greece	City-scale	Open	Mobility, mmWave, cloud computing
PhantomNet [108]	University of Utah, USA	University-wide	Open	Mobility, D2D, eMBB, URLLC, NB-IoT
POWDER [109]	University of Utah Campus, USA	City-scale	Open	Software-defined networks (SDN), massive MIMO
SK Telecom 5G Playground [100]	SK Telecom R&D Center, Bundang, Korea	Not Specified	Close	3D beamforming, massive MIMO, 4K live broadcast system and AR/VR
TitanMIMO-6 [104]	Nutaq, Québec, Canada	Indoor	Not Specified	Massive MIMO
TurboRAN	**University of Oklahoma, Tulsa, USA**	**University-wide & Indoor**	**Open***	**Refer to Table 3.2**

* TurboRAN testbed does not have a remote access interface at the moment. However, it is open to be used by community via physical visit or remote collaborative experimentation.

Table 3.2: A summary of cellular system research areas supported by TurboRAN.

Ultra-Dense Multi-Tier, Multi-Band Networks (UDMN)	
1	User centric RRM, advanced interference management schemes*
2	Advanced Small Cells/ Multi-tier Heterogeneous networks*
Highly Flexible Architectures: Network Orchestration	
3	Artificial Intelligence (AI)-based System-level Automation*
4	Context Aware RAN/ Proactive-SON (P-SON)*
5	Software RAN/Control and Data Plane Split Architecture*
6	Database-Aided Control and Data Split Architecture (D-MUD)*
7	Operation on Unlicensed Bands*
8	Radio Access Network (RAN) Sharing*
9	Network Function Virtualization**
10	Moving Networks**
Large Scale Machine-to-Machine (M2M) Communications	
11	Advanced Multiple Access Schemes*
12	Device-to-Device Communication (D2D)*
13	Very low power consumption operation modes (IoT)*
PHY Focused Technologies	
14	System-level performance (capacity, EE, QoS) evaluation of mmWave**
15	System-level performance evaluation with new waveforms**

* Indicates a research topic that is fully supported.
** Indicates partially supported research areas but will eventually be captured with future upgrades and expansions.

3.3.1 TurboRAN Overview

TurboRAN is an end-to-end 5G&B network testbed facility designed to facilitate the evaluation of early real-world use case deployment. Unlike many 5G testbeds that allow experiments on the physical, link, and MAC layers only TurboRAN is designed to enable system-level research on the next-generation mobile network as identified in Table 3.2. Located at the University of Oklahoma (OU)-Tulsa campus and managed by AI4Networks Research Center [110], TurboRAN is a university-wide mobile cellular network with a combined indoor and outdoor cell deployment that covers a 300,000 m^2 area. TurboRAN's BS density deployment translates to roughly 57 BS/km^2 with 17 distinct cell sectors, which is in line with the predicted 5G BS density of 40-50 BS/km^2 [111]. Fig. 3.2 provides a schematic of the TurboRAN

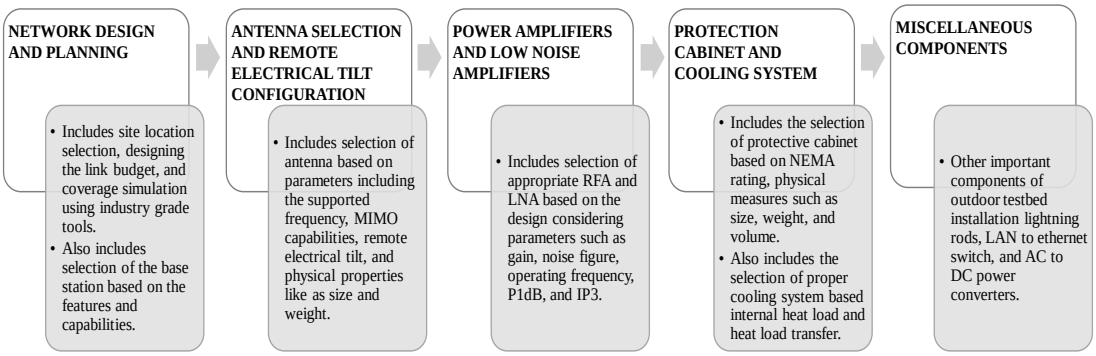


Fig. 3.1: Summary of the steps considered for equipment related design choices and considerations.

network design. The outdoor environment contains roads, pedestrian areas, lawns, trees, and parking lots, representing a typical suburban area. The outdoor cell deployment has 6 microcells, each covering the portions of the campus outdoor area. TurboRAN cells operate in the unlicensed bands below 6 GHz — including ISM – 915.0 ± 13 MHz, ISM – 2450.0 ± 50 MHz, and ISM – 5.8 ± 0.075 GHz, as summarized in Table 3.3. Since these bands lie close to the cellular bands of 800 MHz, 2500 MHZ, and the 5G bands in 5 GHz, results generated by TurboRAN can be applicable to commercial bands. In the near future, TurboRAN will also operate in Citizens Broadband Radio Service (CBRS) 3.5 GHz, upon obtaining an experimental license, in addition to planned mmWave band operation.

Each cell of the TurboRAN has 4 antenna elements forming an array for MIMO and 3D beamforming capabilities. Cell radii in all cells are adjustable through adaptation of the transmit power and antenna tilts. Precisely controlled mobility in the indoor and outdoor can be achieved using unmanned ground vehicles (UGVs) and UAVs, respectively. The TurboRAN testbed has commercial and open source 5G Core (5GCs)/EPCs. As another distinct feature, the testbed incorporates a Hadoop cluster for big data processing. This cluster is connected to 5GCs and EPCs, as well as access points (APs) for applying machine-learning algorithms, allowing for zero-touch automation and proactive self-organizing network (P-SON) functionalities. High-capacity optical fiber cables connect the cells to the 5GCs/EPCs and

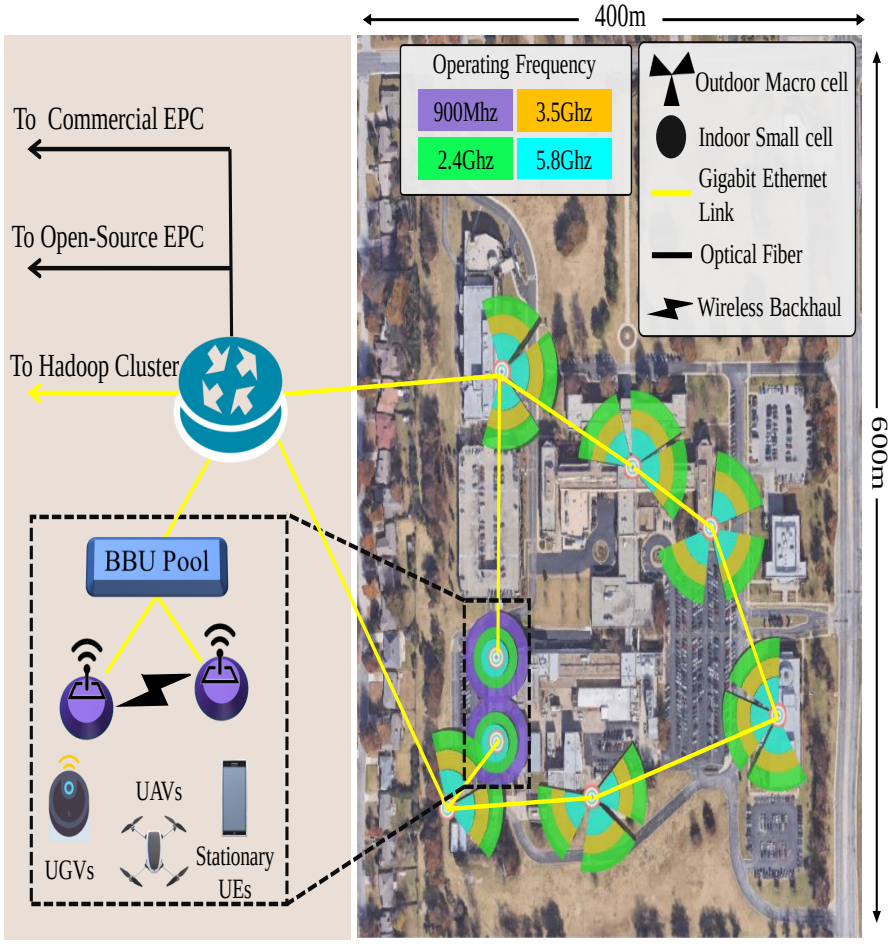


Fig. 3.2: TurboRAN deployment illustration.

large data processing cluster, allowing for data rates of up to 10 Gbps.

Table 3.3: ISM frequency bands supported by TurboRAN.

Frequency Range	Bandwidth	Center Frequency
902 - 928 MHz	26 MHz	915 MHz
2400 - 2500 MHz	100 MHz	2450 MHz
5725 - 5875 MHz	150 MHz	5800 MHz

3.3.2 Network Design and Planning

In each of the TurboRAN outdoor and indoor tiers, an AP consists of three main components: antennas, a software-defined radio (SDR), and a computer node to host the open source LTE and 5G protocol stack. An extensive survey of a large

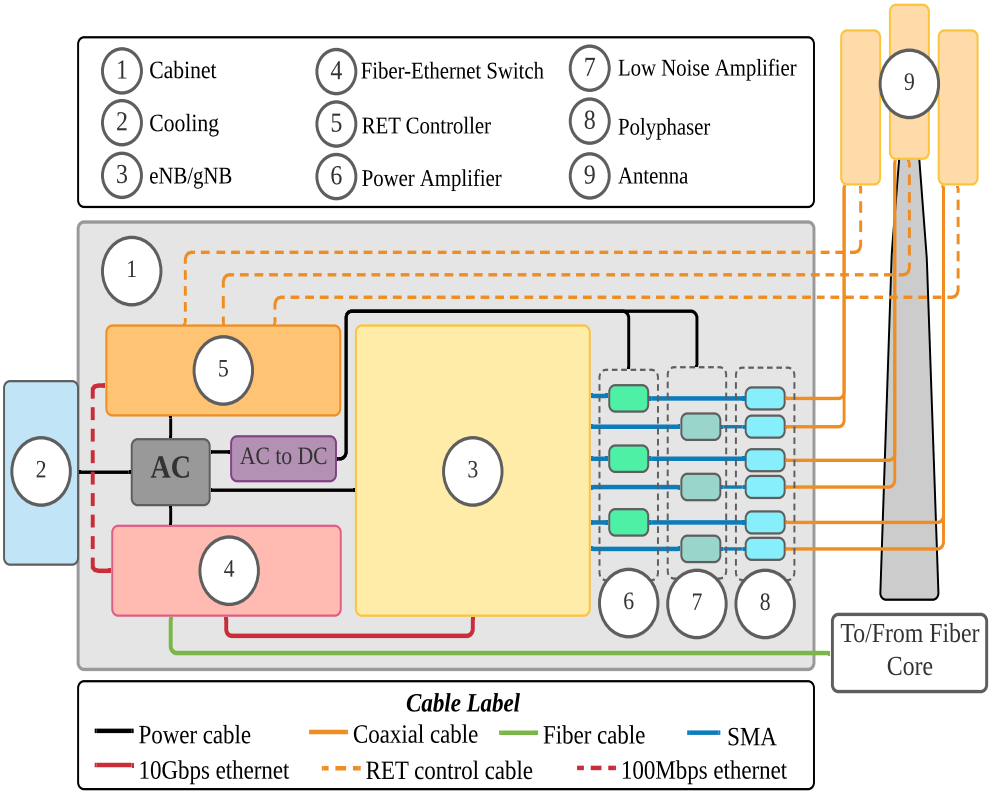


Fig. 3.3: Block diagram representation of the TurboRAN testbed deployment.

number of the SDRs, compute nodes, antennas, chassis, connecting mediums, and open source stacks have been conducted to select the optimal available components to build the APs that meet TurboRAN requirements for each of its planned tiers. While selecting hardware for APs, we also conducted a survey of all potential vendors (e.g., Ettus, NI, Texas Instrument, PRISMTECH, Nuand, AVENET, EPIQ, Sidekiq, Hack RF blue, Nutaq, BeeCube) with the following considerations: 1) cost; 2) bandwidth of the channel; 3) frequency range; 4) MIMO support; 5) clock speed; 6) GPS-based precise clock synchronization; 7) ease of programmability; 8) maximum output power; 9) 10 Gbps high-speed connectivity, and; 10) compatibility with programming languages of the open source stacks.

Fig. 3.3 shows the high-level block diagram used for TurboRAN deployment. This block diagram shows the main components of TurboRAN including base stations, amplifiers, Remote Electrical Tilt (RET) controller, switch, cabin, cooling, and

antenna system. Additionally, this block diagram shows the connection between the different components as well as the types of cables used in the deployment. The protective cabin shown in ① encloses the majority of the equipment. To maintain the temperature inside this cabinet, a cooling system shown as ② is used. Labeled ③ in the figure is the main equipment composed of SDRs and 3GPP compliant software to act as the base station. To connect it to the Internet and other base stations via fiber, a fiber-ethernet switch is exploited as shown in ④. The SDRs of the callbox are connected to the amplifiers using SMA cables. More specifically, the transmitter is connected to RF power amplifiers (PA) ⑥ while the receive side is connected to low noise amplifiers (LNA) ⑦. Before terminating the coaxial cables to the antenna ⑨, a surge protector ⑧ is inserted to protect the base stations against lightning strikes. Finally, to regulate the tilt adjustment remotely, a RET controller ⑤ is attached to the antenna through a RET controller cable. Most of the components use an AC power supply except for the amplifiers which need DC. For amplifiers, we use AC to DC converter.

Network Planning

Network planning is one of the most critical parts of testbed deployment. At a high-level viewpoint, network planning involves processes such as base station location selection, link-budget analysis, and coverage simulation including transmit power, tilt, and azimuth optimization.

In selecting the base station location, we considered several factors. First, the location should be easily accessible not only to avoid any problem during the installation and maintenance but also for manual reconfiguration of the site if needed for future experiments. Second, the availability of facilities such as power source and fiber connectivity has certain merits like reduced efforts for setup, and cost-effectiveness. We identify six outdoor and two indoor locations that are ideal to deploy TurboRAN

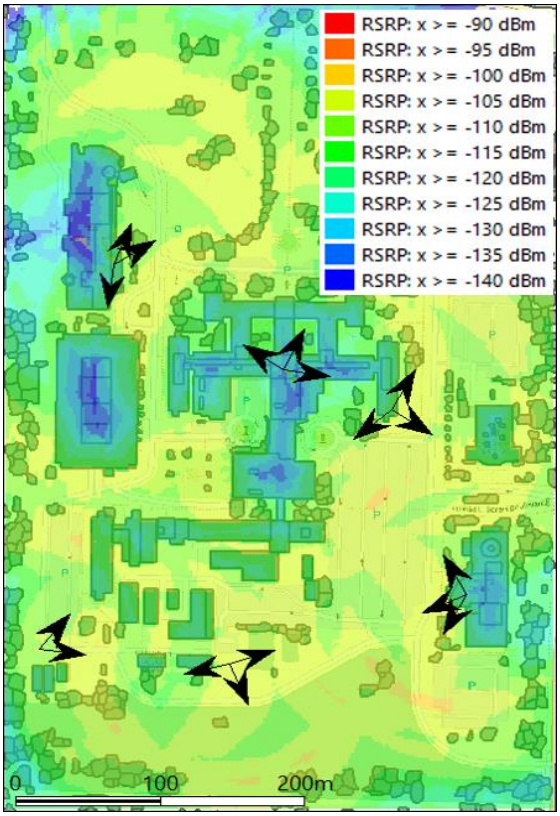


Fig. 3.4: Optimized RSRP simulations using Atoll.

core sites as shown in Fig 3.2. After determining the site locations, the next logical step is to conduct a link-budget analysis, which entails determining the gains and losses experienced by the communication signal as it travels from the transmitter to the receiver. The link-budget ensures that data is transmitted intelligibly with a reasonable signal-to-noise ratio. Finally, the validation process is performed by leveraging tools such as network simulators that are built for coverage calculations. We use Atoll [86], a ray-tracing based industry leading radio network planning software, to generate a practical coverage map for TurboRAN deployment. TurboRAN's coverage simulations take into account important environmental characteristics as geography, clutter type, building heights, and vegetation. Moreover, we leverage the automatic cell planning (ACP) feature of Atoll to optimize the critical base station parameters (tilt and azimuth). Fig. 3.4 shows the Reference Signal Received Power (RSRP) plot after performing the tilt and azimuth optimization process.

Base Stations

Though the core network and antenna system are vital components, the base station is still considered as the heart of the testbed. This is primarily because of the base station software support and the hardware capabilities that dictate the use-cases we can perform in our deployed testbed. In choosing a base station for TurboRAN, the availability, affordability, and functionality of software that supports 4G and 5G protocol stacks are among the most important factors to take into account. Often time, open source software provides more freedom compared to close source or proprietary software with regards to configuring different test scenarios. Similarly, the intended coverage of the network (i.e., indoor or outdoor) and the type of base station (i.e., macro cell or micro cell), are also integral parts of the selection process. Support for 3GPP standard features including MIMO, carrier aggregation, multi-technology connection, and mobility or handover should also be considered. The feature selection of the SDR that comes along with the base station is also another key consideration aspect for the base station selection. Features including downlink (DL) and uplink (UL) antenna ports count, maximum radiated power, maximum allowable bandwidth, and operating frequencies, should be considered. Lastly, software support, troubleshooting, and upgrades are also important. Table 3.4 shows the comparison between some of the 5G base stations we considered for TurboRAN. Among other solution providers, we shortlisted the three most suitable options including Anritsu MT8000A [112], Ixia Keysight [113], and Amarisoft Callbox [114]. Anritsu MT8000A offers a wide range of features such as 5G standalone (SA) RAN, 5G core, and support for mmWave. However, most of these functions and features need to be purchased individually which makes this option uneconomical and complicated to handle. Furthermore, MT8000A only supports tests and analysis of Layers 1 and 2, limiting the experimentation that can be performed. Meanwhile, Ixia Keysight has most of the vital features already

included in the standard package. However, it does not support mmWave and does not include 5GC implementation.

For a testbed capable of testing various innovative AI-based zero-touch automation solutions and Self-Organizing Network (SON) solutions, there is a need for a base station with flexible software support. For this reason, we chose the Amarisoft Call-Box series equipment for TurboRAN [115]. Amarisoft Callbox is a 3GPP-compliant base station and core network that can be used for functional and performance testing. It is powered by a deployment-ready software suite that allows modification for additional flexibility and experimentation. We determine Amarisoft Callbox to be the best solution for TurboRAN because of its vast range of supported technologies and functions.

We use two different series of the Amarisot Callbox namely Amarisoft Callbox Pro and Callbox Classic. Table 3.5 summarizes the key features of Amarisoft Callbox Pro and Classic. Amarisoft Callbox works with a variety of networks, including 5G Standalone (SA), 5G Non-Standalone (NSA), LTE, and NB-IoT. The 4G and 5G implementations are compliant with 3GPP-release 14 and 15, respectively. In comparison to other alternatives, EPC implementation is combined with 3GPP-compliant 5GC. It supports both Time Division Duplex (TDD) and Frequency Division Duplex (FDD) modes of transmission, with a supported frequency band of 500 MHz to 6 GHz along with the bandwidth of 200 kHz to 56 MHz, and the maximum downlink and uplink throughput reported are 1200 Mbps and 150 Mbps, respectively. Further, it can support up to 6 SDRs for Amarisoft Callbox Pro and 3 SDRs for Amarisoft Callbox Classic, offering support for up to 1000 active UEs distributed within one cell.

Table 3.4: Comparison between select 5G network testbed solutions.

5G Solution Provider	Supported Technology	Core	Frequency	MIMO	Carrier Agg.	Supported Layers for Analysis	Possible Test Scenarios	Limitations	Ref.
Anritsu MT8000A	5G SA with option of 4G	EPC and 5GC for additional cost	All FDD and TDD bands in sub-6 GHz and mmWave	4x4	8 CA	L1 and L2	-Dynamic Spectrum Sharing (DSS) -Throughput Tests - Handover Tests	-Most functions and features need to be purchased individually. -Supports only L1 and L2.	[113]
Ixia Keysight	5G SA, 5G NSA	EPC	All FDD and TDD bands in sub-6 GHz	4x4	4 CA	All Layers	-Capacity Tests -Throughput Tests -Voice and video quality tests -Mobility Tests	Support for 5GC and mmWave are not available.	[114]
Amarisoft Callbox	5G SA, 5G NSA, and 4G	EPC and 5GC inclusive	All FDD and TDD bands in sub-6 GHz	4x4	8 CA	All Layers	-Throughput Tests -Handover Tests -VoLTE Tests -NB-IoT Tests -Capacity Tests	Support for mmWave is not available.	[115]

3.3.3 Antenna Selection and Remote Electrical Tilt Configuration

In addition to base station selection, one of the most challenging tasks for an over-the-air testbed deployment is the choice of the best antenna. Although the Amarisoft Callbox includes antennas, they are simple dipole low-gain antennas with a limited range meant for indoor use. Clearly, these antennas do not meet the requirement of the outdoor deployment required in TurboRAN. More appropriate outdoor antennas are required to provide wider coverage. Based on the anticipated

Table 3.5: Key features of Amarisoft CallBox.

Feature	**Callbox Pro**	**Callbox Classic**
RAN	4G, 5G (NSA, SA)	
LTE-M/NB-IoT	✓	
Core	EPC/5GC	
Duplex	FDD/TDD	
#SDRs	6	3
#Cells	6	3
IMS Server	✓	
eMBMS Server	✓	
GPS synchronizer	✓	
OS	Linux	
CPU	Inter Core i9	Inter Core i7
3GPP Release No.	Release 14 for 4G / Release 15 for 5G	
Supported Frequency	500 MHz - 6 GHz	
Supported Bandwidth	200 kHz - 56 MHz	
Indoor Wireless Range	10 m	
#UEs	1000	
UE Category	0 - 12	0 - 10
Maximum DL Throughput	1200 Mbps	600Mbps
DL Modulation	256QAM	
Maximum UL Throughput	150 Mbps	150 Mbps
UL Modulation	64QAM	
CA	8 CA	
MIMO	4x4	2x2
Handover	Intra Base Station & Inter Base Station	

deployment, we determine the appropriate antenna with the required properties. Antenna selection for TurboRAN is influenced by factors such as support for ISM bands, MIMO, remote electrical tilt, and physical properties like weight and size.

TurboRAN is designed to initially operate on the ISM band with configurable SDRs. To maximize the SDR capabilities, the antenna should complement the SDRs and support a variety of ISM frequency bands. Due to the space and tower loading constraints, we focused on antennas that can support the frequency range shown in Table 3.3. Although, indoor antennas provided with Amarisoft Callbox support multiple frequencies, finding suitable commercial antennas with higher gain that can support the target frequency is not a straightforward task. Most of the current commercially available antennas operate on commercial bands such as 3.3 GHz to 4.2 GHz. Other antennas that provide support for the ISM band, support only one or two frequency bands at max. There exist antennas that support at least 3 frequency bands such as [116]. However, with the increase of the number of supported frequency bands, there is an increase of antenna elements, that ultimately increase the total weight of the antenna. To utilize the existing poles on the campus without any retrofitting, and avoid installation of the 5m poles, the antennas should weigh light.

Table 3.6: Selected antenna specification.

Parameters	**Values**		
Frequency (MHz)	1695-2690	3400-3800	5150-5925
Port Configuration	4 ports	2 ports	2 ports
Gain (dBi)	14	10	6
Beamwidth (degrees)	65		
Tilt	RET		
Impedance (Ω)	50		
Voltage Standing Wave Ratio	1.5		

Most commercial cellular networks and testbeds incorporate MIMO in their deployment, as MIMO has proven to have several advantages. In the industry, MIMO

is used to increase the throughput of the base station by using multiple antennas transmitting at the same time. For testbeds, having MIMO capability can open more alleys in terms of experimentation. In addition to MIMO, modern cellular networks are equipped with functionality to remotely control the antenna tilt settings, known as RET. This feature has become an indispensable part of the cellular network due to the convenience as well the savings in the operational cost it brings to the network operators. Hence, RET eliminates the need to physically visit the site for tilt adjustments. Therefore, RET and MIMO support were two other criteria considered in the antenna selection for TurboRAN.

After evaluating various antennas that fitted the above-mentioned desirable attributes, we chose the AW3639 antenna from Alpha Wireless [117]. This antenna has 12-ports and can operate on 1695 to 1995 MHz / 1920 to 2170 MHz / 2170 to 2500 MHz / 2500 to 2690 MHz / 3400 to 3800 MHz / 5150 to 5925 MHz frequencies. It can also support up to 4x4 MIMO. It has a compact design, only 0.61 m in length, and lightweight weighing only 5.9 kg. Table 3.6 shows the configuration of the antenna chosen for the TurboRAN deployment.

Alpha Wireless AW3639 antenna is also AISG 2.0 compatible, i.e., it can support RET controller from other manufacturers abiding AISG 2.0 standard with a range of 2° to 10°, as for TurboRAN, we have incorporated RET. Several RET controllers that are available in the market are considered including the solutions offered by Comba [118] and Commscope [119]. Consequently, Kathrien Central Control Unit (CCU) [120] is chosen as the preferred solution considering various advantages over other options. These advantages include interoperability with most antenna and support for a dual power supply. The CCU acts as an interface between the antennas' remote-control unit (RCU) and the control system, and can be operated both locally and remotely. Fig. 3.5 shows the two RET configuration options, we have considered in TurboRAN deployment. The first option we have considered is

deploying the RET using the daisy chain method. In this method, RET ports of the antenna are cascaded and only one cable is terminated at the RET controller. Even though this method is more cost-efficient, it is prone to complete RET failure due to the existence of a single point of failure. Therefore, for TurboRAN, we decide to use the second approach which is to use a dedicated RET controller cable for each antenna. Although this approach needs more cables, it is more robust against total RET failures.

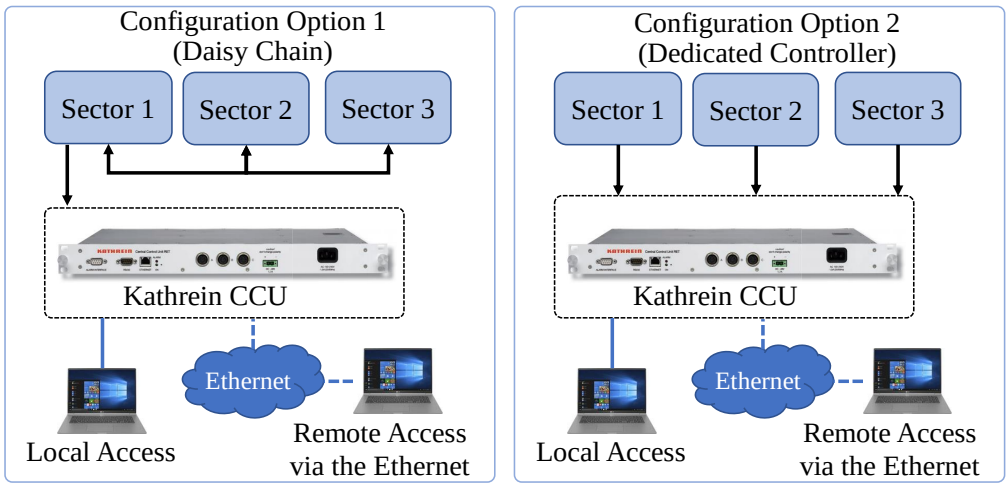


Fig. 3.5: Two options for RET configurations.

3.3.4 Power Amplifiers and Low Noise Amplifiers

The Amarisoft Callbox SDRs can only output 5dBm of signal power at their maximum. This presents a challenge because such a small amount of power can only adequately cover limited indoor regions. Furthermore, this amount of power is insufficient to fuel the external antennas. To address this problem, we use radio frequency amplifiers (RFAs) on the transmit side to increase the signal power originating from the SDRs.

Table 3.7: Power amplifiers and low noise amplifiers requirements.

Parameter	**RFA and LNA Requirements**
Connector	SMA
Gain	>20 dB
Noise Figure	<6 dB
Operating Frequency	(2.4, 3.5, 5.8) GHz
P1dB	>5 dBm
IP3	>15 dBm
Impedance	50

Amplifiers measure the amount of boosting or amplification by the parameter called gain. According to the simulations using Atoll as shown in Fig. 3.4, at least 20dB gain is necessary to ensure coverage for TurboRAN. Another important RFA parameter is the linearity measured by the 1 dB compression point (P1dB). The output power level gain deviates 1 dB less than the linear constant value at the P1dB point. Since the maximum power from the SDRs is 5 dBm, amplifiers with more than 5 dB P1dB can work effectively. Upon reaching P1dB, the amplifier losses linearity and starts producing byproducts such as distortion and harmonics. These harmonics, especially the third-order product, can cause interference to the first-order or fundamental power. Thus, another critical parameter to consider in amplifier selection is known as the third-order intercept or IP3. This parameter is an imaginary point wherein the fundamental power and the third-order power both have the same value. Practically, this point can never be reached as the amplifiers saturate before reaching this point. Usually, IP3 is 10 dB greater than P1dB. Therefore, for TurboRAN deployment, we require an amplifier with IP3 of at least 15 dBm. Table 3.7 shows the RFA requirements for TurboRAN.

The extremely weak signal received by the base station antennas, similar to the transmit side, requires amplification before being analyzed by the SDRs. While providing amplification to the received signal is straightforward, the real challenge, in this case, is the introduction of noise from the amplifier which can compromise the noise level in the extremely weak signal. To address this issue, a specific type

of amplifier called Low Noise Amplifier (LNA) is used to minimize the added noise from the amplifier by utilizing components and topologies that generate less noise. Similar to RFA, the critical parameters for consideration while selecting LNAs are gain, P1dB, and IP3. Noise figure or the sensitivity of the amplifiers to noise is another crucial parameter for LNA. Table 3.7 also summarizes the LNA requirements for TurboRAN.

3.3.5 Protection Cabinet and Cooling System

As previously mentioned, one major limitation of Amarisoft Callbox is not having the support of outdoor deployment. However, we can counter this shortcoming by installing the equipment inside a protective cabinet having the optimal weather control and rain/sleet cover. To allow the TurboRAN to operate in an outdoor scenario, a protection cabinet is another aspect that needs attention. Since Tulsa, Oklahoma features both sub-zero as well as fairly hot days maintaining acceptable temperature in the outdoor cabinet sheltering the equipment is another challenge for outdoor deployment. In deploying TurboRAN, we have considered several cabinets which can provide the needed weather protection for the base station equipment. However, choosing the right cabin for this fragile equipment requires consideration of different factors. In this subsection, we discuss the factors we have considered in choosing the protection cage for the base stations.

Protection Cabinet

NEMA rating is perhaps the most important and first factor to consider in choosing the right enclosure. NEMA which stands for National Electrical Manufacturers Association, is an organization that oversees the crafting of technical standards and specifications for electrical and medical imaging device manufacturing in the United States. The NEMA rating system is a classification system for the types

Table 3.8: Comparison of specific applications of enclosures for outdoor nonhazardous locations (from NEMA 250-2003).

Provides a Degree of Protection Against:	Types of Enclosures									
	3	3X	3R*	3RX*	3S	3SX	4	4X	6	6P
Access to hazardous parts	X	X	X	X	X	X	X	X	X	X
Ingress of water (Rain, snow, and sleet**)	X	X	X	X	X	X	X	X	X	X
Sleet***	-	-	-	-	X	X	-	-	-	-
Ingress of solid foreign objects (Windblown dust, link, fibers)	X	X	-	-	X	X	X	X	X	X
Ingress of water (Hosedown)	-	-	-	-	-	-	X	X	X	X
Corrosive agents	-	X	-	X	-	X	-	X		X
Ingress of Water (Occasional temporary submersion)	-	-	-	-	-	-	-	-	X	X
Ingress of Water (Occasional prolonged submersion)	-	-	-	-	-	-	-	-		X

* These enclosures may be ventilated.
** External operating mechanisms are not required to be operable when the enclosure is ice covered.
*** External operating mechanisms are operable when the enclosure is ice covered.

of environmental conditions in which a given enclosure can be employed. NEMA ratings for non-hazardous, outdoor locations are shown in Table 3.8. For TurboRAN deployment, we have selected a cabinet that conforms to NEMA rating Type 4. In summary, Type 4 enclosures can be used for both indoor and outdoor locations. It is watertight and dust-tight providing protection against rain, snow, sleet, windblown dust, splashing water, and hose-directed water. However, when the deployment area is near the bodies of water such as sea which can cause corrosion, one should go for Type 4X, while 6 and 6P are recommended if there are chances of flooding and water submersion.

Another factor to consider is the physical measures of the outdoor cabinet such as its size, weight, and volume. First, weight should be considered especially when the cabinet is wall or tower mounted. It should be light enough to be anchored on the walls or pole. In TurboRAN deployment, we have designed our cabinets to be floor mounted wherein a pad is built on the ground and the cabinet is placed on top. With this regard, weight is not the main factor we considered in choosing the cabinet. The more crucial factor to consider is the cabinet size and volume. The cabinet should be able to enclose not only the Amarisoft callbox but also the other equipment that are needed to make the TurboRAN up and running (i.e., amplifiers, switch, RET controller, etc.). In addition, future expansion should also

be considered in choosing the best size of the outdoor enclosure.

After consideration of the above-mentioned criteria (i.e., NEMA rating, weight, size, and volume), we shortlisted two potential candidates for further evaluation. First is the OD-30DXC, a 15 Rack Unit (RU) enclosure used in industrial applications in the field of telecommunications, fiber optics, military, and public safety. It would have been a perfect fit for TurboRAN deployment. However, if future expansion is considered, this particular design does not have room for the installation of any further equipment beyond what is already needed for TurboRAN's current deployment. Thus, we made a decision to use a larger enclosure called OD-50DXC. This 26 RU outdoor enclosure is almost double the size of the OD-30DXC giving enough room for future expansion and equipment addition. This cabinet, which supports NEMA Type 4 and 4R configurations, is utilized in applications similar to the OD-30DXC. It also includes a side spool-up cabinet for cable entry and an electrical panel with internal outlets, both of which are missing from the OD-30DXC. Table 3.9 provides the specification summary of OD-50DXC.

Table 3.9: Outdoor enclosure specification.

NEMA Rating	NEMA 4 and 4X configurable
Exterior Enclosure Dimensions	51"H x 28"W x 30"D
Approximate Shipping Weight	73 kg
Size in RU	26 RU
Mounting Type	Floor Mounted

Cooling System

The temperature in Oklahoma can easily cross 37.5 °C in summer. Inside a metallic enclosure the temperature can easily rise above the outdoor temperature. This can lead to temperature high enough that can cause severe damage to the base station equipment despite the protection offered by the OD-50DXC cabinet. That is why TurboRAN outdoor enclosures must have a cooling system to keep the temperature

within the equipment's operating range. Due to the harsh summer conditions in Oklahoma and the delicateness of the equipment, cooling using the built-in enclosure fan would not suffice. To address this challenge, we have added an air conditioning system to the TurboRAN outdoor deployment unit designs to maintain the temperature inside the cabinet.

To select the proper size of the air conditioning to be used, worst-case conditions need to be considered. Internal heat load and heat load transfer are two elements to consider while selecting the appropriate cooling system for the enclosure. The former refers to the heat load generated by the equipment components inside the cabinet, while the latter refers to the heat loss or gain from the ambient air that enters the enclosure through its walls. Internal heat load expressed in British thermal unit per hour (BTU/H) can be calculated using the maximum heat output specifications of the equipment expressed in Watts. Conversion of Watts to BTU/H is given as:

$$\phi = \omega \times 3.413 \tag{3.1}$$

where ϕ is the internal heat load (IHL) in BTU/H and ω is the maximum heat output specifications of the equipment expressed in Watts. The sum of all the internal heat loads of all the components inside the cabinet is the total internal heat load. For TurboRAN deployment, Table 3.10 shows the maximum heat output of all the equipment inside the cabinet equating to a total of 518 W. Assuming that around 99% will be converted into heat (i.e., 512 W), gives an IHL of 1,747 BTU/H. Meanwhile, heat load transfer (HLT) can be calculated using:

$$\vartheta = 1.25 \times S \times (\tau_a - \tau_i) \tag{3.2}$$

where ϑ is the HLT in BTU/H, S is the enclosure surface area in ft^2, τ_a is the maximum outside ambient air temperature in °F and τ_i is the maximum allowable

internal enclosure temperature in °F. The constant value of 1.25 is an industry standard for metal enclosures. We calculate HLT based on S of 4.46 m^2 (48 ft^2), τ_a of 37.5 °C (100 °F) and τ_i of 26.5 °C (80 °F) to be 1,200 BTU/H. The total cooling capacity (CC) of the air conditioning system is the sum of IHL and HLT. IHL of 1,747 BTU/H and HLT of 1200 BTU/H brings the total CC to 2,947 BTU/H. Thus, we have decided to use a 3,000 BTU/H cooling system to meet this requirement.

Table 3.10: Maximum heat output of the equipment inside the cabinet.

Equipment	**Maximum heat output**
Amarisoft Callbox	200 Watts
RET Controller	50 Watts
Switch	250 Watts
Amplifiers	1.5 Watts (x 12)
Total	518 Watts

3.3.6 Miscellaneous TurboRAN Components

Other important components of TurboRAN's outdoor installation include a surge protector or lightning surge suppressor to protect the equipment from electromagnetic pulse (EMP) caused by lightning strikes or other strong electrical changes. Additionally, lightning rods are installed on the antenna poles to protect the antenna from lightning strikes. The lightning rods as well as the surge protector are terminated to a copper bus bar which goes directly to the ground. Meanwhile, LAN to ethernet switch is used to connect the Amarisoft Callbox to the LAN of the campus and the internet. Lastly, to provide DC power to the amplifiers, multi-port AC to DC power converters are utilized.

3.4 Case Study: Mobility-based Inter-frequency Handover

After the successful deployment of a testbed, it is important to verify its operation functionality. One approach to achieve this is to design and test common network scenarios, then evaluate the results. We consider the following criteria in selecting

the suitable use case to present: 1) the use case should require and validate over-the-air capabilities of the testbed; 2) the use case should address a problem that is not yet covered in literature; 3) the use case should address a practical problem that is relevant for industry; and 4) the use case should be simple to understand for researchers with no prior experience of testbed. Out of plethora of use cases TurboRAN can support, we include a mobility management-related use case since it fits all of aforementioned criteria. Additionally, we chose this use case since mobility management is one of the most challenging tasks in cellular networks, and few testbeds support mobility functions.

In 5G cellular networks, to cater to the increasing data rate requirements, dense networks are deployed. For the provision of uninterrupted and efficient communication services to mobile users in emerging dense networks, handover is an essential enabling feature. In addition, to improve the coverage range, Quality of Service (QoS), and overall data rate, generally heterogeneous networks are deployed. However, despite the auspicious aspects of such networks, the small cells in a dense network increase the handover attempts for a mobile user. Hence, the issues related to mobility such as ping-pong, early and late handovers, and handover failures become more challenging problems in the emerging ultra-dense multi-band multi-tier networks such as 5G [121]. These issues, if not taken care of in a timely manner, can lead to degradation in Key Performance Indicators (KPIs) including data rates, latency, user Quality of Experience (QoE), power consumption of communication devices, and signaling overhead. Thus, the cellular networks must be equipped with efficient mobility management systems to avoid degradation in the aforementioned KPIs. Not many simulators model HO and mobility with realistic details. Analytical models on the other hand often omit mobility altogether to have tractable results and are asymptotic. Testbeds are the platforms that offer the most resource capabilities to test mobility management capabilities and algorithms. Therefore, in this section, study TurboRAN's potential to investigate mobility management

realistically. In the process, we demonstrate how improper configuration of the handover-related parameters can have a detrimental effect on ping pong and performance in a real network. We demonstrate the significant influence especially to inter-frequency handovers, which occur between the cells of various frequency layers.

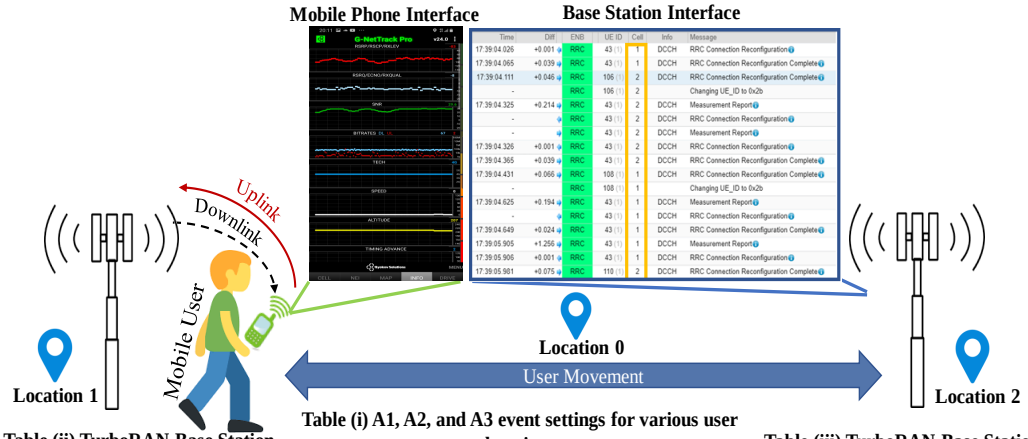


Table (ii) TurboRAN Base Station

Base Station # 1 Parameters	Values
Operating Frequency Band	Band 7
Tx Power	5 dBm
Cell ID	0x01
DL center Frequency	2655 MHz

Table (i) A1, A2, and A3 event settings for various user locations

Handover Parameters	Experiment 1	Experiment 2	Experiment 3
A1 RSRP (dBm)	-80	-80	-105
A1 TTT (ms)	128	256	640
A2 RSRP (dBm)	-80	-80	-110
A2 TTT (ms)	128	256	640
A3 offset (dB)	-2	0	6
A3 TTT (ms)	128	256	640

Table (iii) TurboRAN Base Station

Base Station # 2 Parameters	Values
Operating Frequency Band	Band 4
Tx Power	5 dBm
Cell ID	0x02
DL center Frequency	2130 MHz

Fig. 3.6: Inter-frequency handover use case experimental setup.

Fig. 3.6 shows the setup for the experiment using the TurboRAN base station (Amarisoft Callbox). The SDRs are programmed to act as base stations that operate on various frequencies. We considered the default frequency settings of the TurboRAN base station, i. e., the first SDR is set to operate in Band 7 with 2655 MHz center frequency while the second SDR is set to operate in Band 4 with 2130 MHz center frequency. The maximum transmit power of both SDRs is 5 dBm. The summary of base station parameters is shown in Tables (ii) and (iii) of Fig. 3.6. To analyze the effect of handover under different configuration settings, a mobile user moves out of the coverage of one base station towards the coverage region of another base station, thus causing a handover (HO).

We perform several experiments using different settings of HO-related parameters such as events A1, A2, and A3. These events that are used to trigger handovers are also summarized in Table 3.11. For these experiments, we use RSRP-based HO triggering. When the serving base station's RSRP falls below a threshold, event A2 is triggered. This event is used to start the measurement gap or time, during which the user measures the signal condition of other frequency layers. When the RSRP of the neighboring base station exceeds that of the serving base station, Event A3 is triggered, and the handover is performed. Meanwhile, event A1 is communicated to the serving base station to cancel the measurement gap if no appropriate cell is detected during this period and the serving base station's RSRP improves above a threshold. These events are triggered if the condition is sustained for the duration specified by the time-to-trigger (TTT) parameter.

For experiment 1, a set of parameters are configured in the TurboRAN base station (Amarisoft software terminal), such that the handover from one base station to the other becomes very easy. To trigger the measurement gap quickly, we set A2-threshold and A2-TTT to -80 dB and 128 ms, respectively. Furthermore, A3-offset is adjusted to a negative value (-2 dB), causing handovers to occur even if the RSRP of the target base station is lower than that of the source. Additionally, A3-TTT is set to a small value (128 ms) for faster triggering. Meanwhile, parameter settings in experiment 2 make the handover moderately difficult compared to the parameter set for experiment 1. For the second experiment, we use 0 dB for A3-offset to force the handovers to happen when the source and target base station RSRPs are equal. Additionally, compared to experiment 1, we use a longer A3-TTT of 256 ms to delay the handover triggering. Lastly, for the third experiment, the set of parameters are configured such that the handover between base station 1 and 2 are avoided. To do this, we set A2-threshold to a low value equal to -110 dB to minimize measurement gaps. Then, we set the parameter A3 offset to a high value of 6 dB, in addition to setting a longer TTT of 640 ms. For this final experiment,

we set the event A1 threshold to a low value (-105 dBm) so that the measurement gaps are canceled easily, hence, making sure the handover will not happen. The summary of the events settings for different experiments is given in Table (i) of Fig. 3.6.

Table 3.11: Summary of intra-RAT measurement events for LTE and 5G NR.

Event	Event Description	Function
A1	Serving base station becomes better than threshold	Cancel measurement gap
A2	Serving base station becomes worse than threshold	Start measurement gap
A3	Neighboring base station becomes offset better than serving base station	Initiate handover

Fig. 3.7 demonstrates the HO results for the three experimental settings in the TurboRAN and the serving time for different duration of times (i.e., 180, 360, and 540 seconds). A mobile user moves within the coverage region of the two TurboRAN base stations for the three aforementioned time durations. A mobile application (G-NetTrack Pro), shown at the top of Fig. 3.6, is used to track the number of handovers and serving time. On one hand, it can be observed through Fig. 3.7 that many handovers are also deteriorating for the stable service provision, causing ping-pong effect. On the other hand, if we make the parameter set to be highly stringent, handovers are suppressed even in cases where the user's RSRP value is very low and the desired data rates are not delivered. This captures the fact that late handovers also have a negative impact on the network performance. While observing the serving time for the three experiments, this time is much higher for the third experiment with stringent parameter settings.

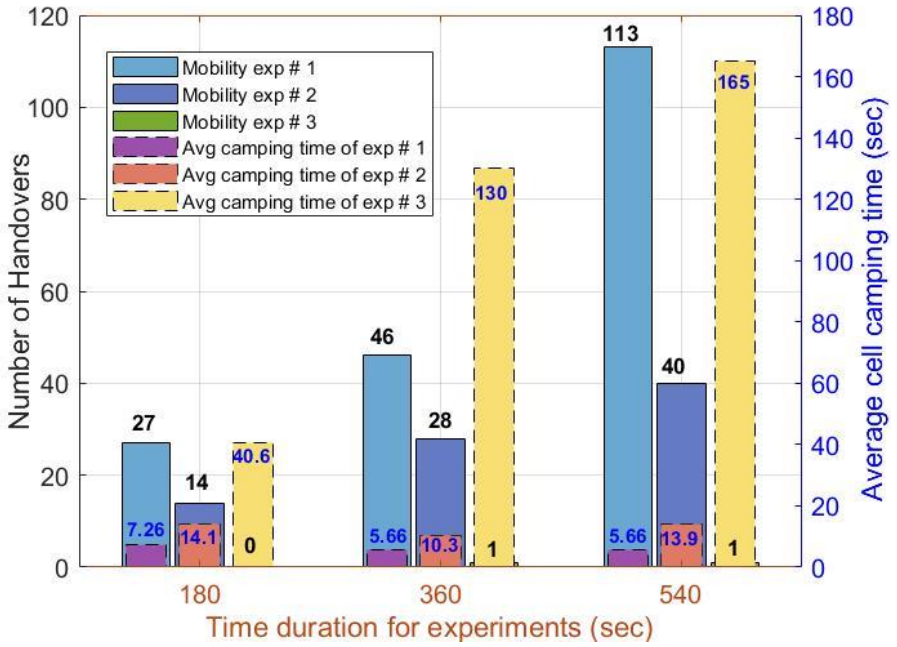


Fig. 3.7: Number of handovers under mobile user settings.

Due to factors such as shadowing, different events configurations can cause the HOs even if the user is stationary, lead to ping-pong effect. The number of HOs while a user is static and its impact on the average RSRP is depicted in Fig. 6.2. For this use case, we investigated three scenarios: 1) the mobile is in the center of the two TurboRAN base stations, 2) the mobile is near the TurboRAN base station 1, and 3) the mobile is near the TurboRAN base station 2. The settings for experiment 1 (default settings) generate ping-pong effect with the maximum number of handovers, even when the user is not moving. For the static case in experiment 3, no handover is detected. In addition, as the setting for experiment 2 are the best suited for the given scenario, i.e., the occurrence of handover happens only when the serving RSRP is not satisfactory, hence, keeping the data rates consistent for the user and providing satisfactory QoS.

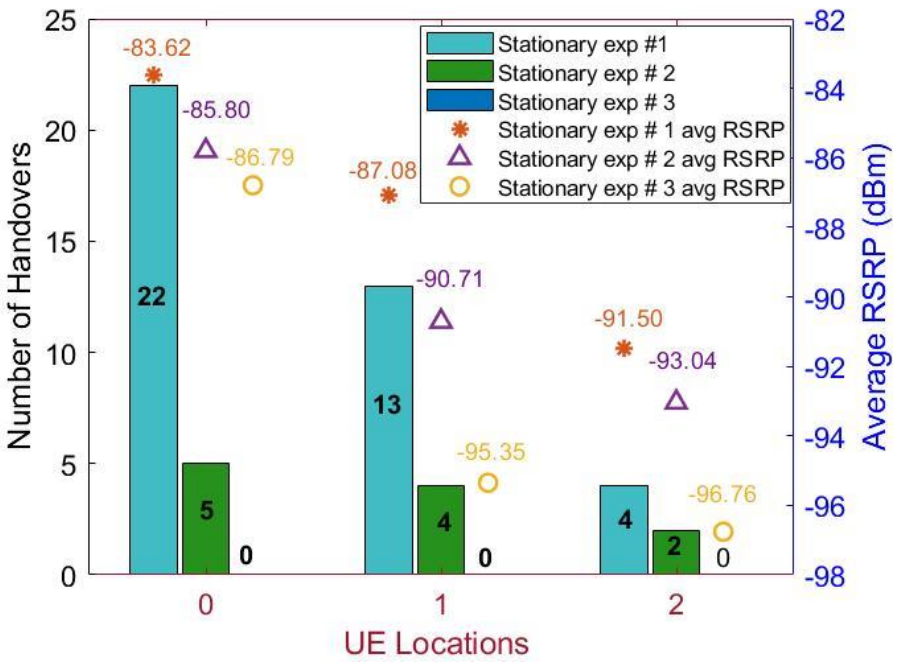


Fig. 3.8: Number of handovers under static user settings.

3.5 Conclusion

Given the complexity of cellular networks and the number of components involved in an end-to-end network, designing and deploying a system-level cellular testbed can be challenging and overwhelming. To address this issue, leveraging insights from our hands-on experience of deploying the TurboRAN testbed, this chapter presents in detail the design considerations and deployment challenges one might face when implementing a 5G and beyond network testbed. We discuss the challenges and approach taken in the selection of all main components of the testbed. This chapter also present the rationale behind using the selected hardware and software solutions for the base station as well as their limitations such as limited indoor coverage offered by most testbed base station vendors. This chapter then explain how these limitations can be addressed by using outdoor antennas and amplifiers, to boost the transmit and receive power. Additionally, we elaborate the important features of the protective enclosure and the cooling system to enable the effective outdoor

deployment of the base station. Finally, we demonstrate TurboRAN's functionality through a case study involving the influence of sub-optimal mobility parameter setup on user experience. This case study shows a typical experiment through which TurboRAN testbed can be used to gather realistic insights on system-level performance aspects (e.g., mobility/handover management in multi-cell environment) that are not possible with most existing simulation or analytical models.

Other use cases of TurboRAN include AI-based zero-touch optimization of parameters such as tilt and transmit power, resource allocation and scheduling, propagation model validation, and development of sophisticated multiple access techniques, among others. Validation and evaluation of these use cases through different validation setups, experiments, and results are beyond the scope of this chapter and will be the topic of future dedicated experimental studies.

CHAPTER 4

Addressing Key Challenges in Using AI for Mobility Management and Zero Touch Automation in Emerging Cellular Networks

4.1 Introduction

This chapter aims to develop a framework to enable a zero-touch mobility optimization that can be validated using the computationally efficient simulator discussed in Chapter 2 and the TurboRAN testbed discussed in Chapter 3. The first step towards achieving zero-touch system level optimization is to develop models for COP-KPI relationships. However, the complexity of this task increases as the number of COPs grows. There are currently more than 2500 COPs in one base station of the current 5G network [122]. Fig. 4.1 shows the complex relationship of select COPs and KPIs in a 5G network in the form of a Venn diagram. A typical engineer working on COP-KPI optimization understands the system level relationship between only a handful of the total COPs and KPIs through deep domain knowledge and field experience and are mostly qualitatively. For instance, RAN optimization experts perform qualitative analysis to tune a few COPs such as tilts, azimuth, and transmit power using their in-field training and experience-based domain knowledge. However, this knowledge is mostly restricted to a limited number of COPs and KPIs for a given human expert. In addition, even SOTA SON solutions encompass a minute portion of the total COPs to maximize all KPIs of interest. Moreover, in Fig. 4.1, we present several network KPIs such as mobility, retainability, accessibility and utilization alongside the COPs which impact one or more of these KPIs. As shown in the figure, there are several parameters which influence one or more KPIs. This implicates those changes in these parameters affect several KPIs, either in a

positive or a negative way. Clearly, comprehension and simultaneous optimization of all these complex COP-KPIs relationships is almost impossible through human efforts even with the extensive experience and domain knowledge. Thus, it is fitting to conclude that the probability of current manual optimization even with SOTA SON paradigm in achieving and maintaining "optimal" system level configurations is very low.

The contributions and organization of this chapter can be summarized as follows: We first show the need for an automated COP-KPI solution by showing the complex relationship between several COP-KPI optimization (Section 4.2). We then characterize the challenges in the context of cellular network system-level problems and highlight through examples their impact in limiting the potential of AI in mobile network system-level optimization (Section 4.3). We then present potential solutions and recommendations to address these challenges while discussing the advantage and disadvantages of each solution approach (Section 4.4). By integrating these solutions with an AI-based COP-KPI optimization solution we propose a zero-touch optimization framework for the next-generation cellular network called AZTEK (AI-Enabled Zero-Touch System-Level KPI Optimization) (Section 4.5). The primary objective of AZTEK is to minimize the dependency of the system-level COP-KPI optimization process on human skills and hit-and-trial-based SON solutions by utilizing AI models to their maximum capabilities despite sparsity and scarcity of real training data. We then evaluate the potential of AZTEK to automate a system-level COP-KPI optimization through a use case that traditionally requires deep domain knowledge and manual hit-and-trial-based COP tuning (Section 4.6). In Section 4.7, we present the key conclusions and insights of this work.

4.2 AI as Key Enabler for Zero-Touch Optimization in Emerging Cellular Networks

Artificial Intelligence in the form of Machine learning (ML) or Deep Learning (DL) is a powerful tool that can be used to model and map out complex functions that are not easily interpreted mathematically from data [123]. This capability makes AI a promising tool for quantifying the complex relationship between mobility parameters and network KPIs. For instance, a machine learning model can be trained on COP data related to mobility to learn the behavior of KPIs with changes in these parameters. This model will capture the highly intricate COP-KPI relationship that is otherwise impossible to analyze through traditional mathematical methods. Once this model is trained, it can estimate the value of KPIs with inputs of mobility parameters.

However, a realization of the full potential of AI implementation for the next generation networks is overcast by two major challenges. 1) The sparsity and scarcity of the training data for system-level COP-KPI model training, and 2) the design and optimization of hyperparameters in an AI model to achieve optimal performance on sparse and noisy cellular network system-level COP-KPI data. Unless properly addressed, these challenges will continue to undermine the capabilities of AI and impede the development of a fully automated KPI optimization process keeping zero-touch automation an elusive goal.

4.3 The Two Key Challenges that Debilitate the Use of Off-the-shelf AI in Cellular Networks

In this section, we discuss the two key challenges that impede the full potential of AI in enabling zero-touch network optimization.

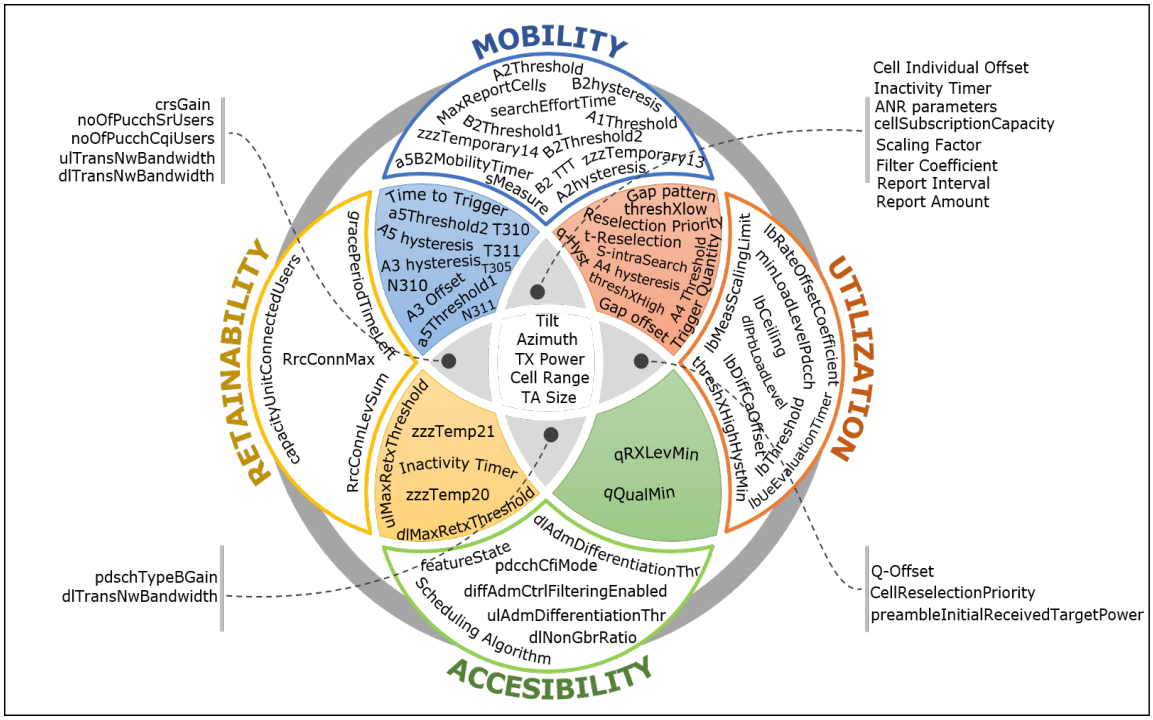


Fig. 4.1: A Venn diagram showing the complex relationship between COPs and KPIs.

4.3.1 Sparse data from a live network

A huge and representative dataset of the COP-KPI relationship is necessary to effectively train an AI model. However, getting these data is still a challenge, particularly in the wireless communication industry. One of the main challenges in collecting data from the live network is the valid reluctance of network operators to experiment on a wide range of COPs in live networks, to avoid poor user experience and churns. In addition, the tendency of the operators to stick with fixed COP values make this sparse data non-representative and unevenly distributed, irrespective of the data size. Moreover, even if the willingness is there, trying all the possible combinations of COPs and observing how KPIs behave in a real network is simply not possible given the vastness of COPs search space and the network size. Furthermore, conducting independent field trials on a large scale is costly and time-consuming. In dynamic scenarios, where the number and locations of measurements constantly change, measurement of the radio frequency field strength values at every point of interest is simply impossible. Finally, mathematical models are

not suitable to generate training data as most of them anchor on too many assumptions and simplifications that are not enough to capture the complexity of a real network. The data-driven models trained on this sparse, unevenly distributed, and non-representative data cannot fully learn the COP-KPI relationship, hence would be sub-optimal and can even result in severely degraded KPIs in some cases.

4.3.2 Hyperparameter optimization

In the realm of ML and DL, hyperparameters are the direct controllers of the behavior and performance of a model. For instance, even though we manage to gather the needed representative data, if the ML/DL hyperparameters are not tuned optimally for a given data or task, the performance of the trained model will be suboptimal. Although some hyperparameters might be safely set to default values, they might still produce suboptimal models. Fig. 4.2 shows some off-the-shelf ML and DL techniques that can be used in zero-touch optimization along with their respective hyperparameters [124]. As shown in this figure, most of the techniques have around a dozen tunable hyperparameters with some, for instance, CatBoost, even having around 46 hyperparameters. While some of these hyperparameters are binary and discrete with fewer than five possible values, the more difficult task is tuning the continuous and discrete parameters with more than five values. Evidently, countless combinations of these hyperparameters are possible, making the search for the optimal one a herculean task. This problem is further aggravated in the case of DL models as the complexity of hyperparameter optimization increases exponentially with the size of the model (i.e., number of hidden layers and neurons per hidden layer).

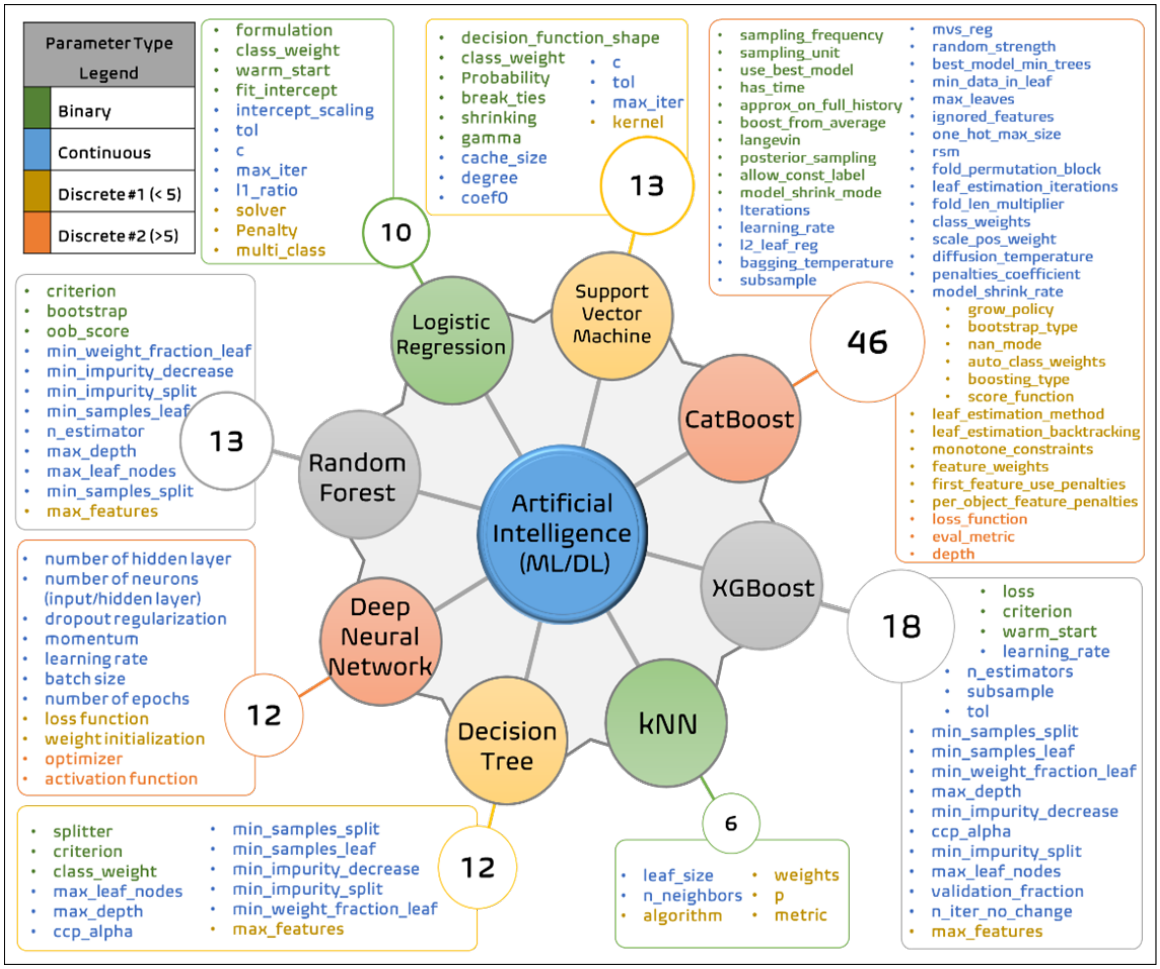


Fig. 4.2: Off-the-shelf ML/DL techniques with corresponding hyperparameters.

4.4 Addressing the Key Challenges in AI for Emerging Cellular Networks

Each challenge requires a different set of approaches that can be leveraged to address them. In this section, we present various techniques that can alleviate these challenges.

4.4.1 Addressing the data sparsity challenge

Although challenging, there exist techniques that can be leveraged to enrich sparse data from a live network through data enrichment.

Data Enrichment Techniques

- Interpolation methods: Different interpolation techniques can resolve the data sparsity challenge if the data is spatially dependent and continuous over space. Among these include inverse distance weighted and its variants, gradient plus inverse distance squared, modified Shepard's method, nearest neighbor, natural neighbor, splines, and kriging [125]. Since the interpolation techniques share the same underlying principle, they can work well when the sparse data is somewhat representative of the whole data or exhibits some degree of spatial correlation. For sparse and non-representative data, these methods are likely to perform poorly even if the data is big. In these cases, the solution lies in either resorting to getting pseudo-real data e.g., through testbeds or generating synthetic data.

- Pseudo-real data through testbeds: The challenge of limited real network data can be in part addressed by generating rich and representative pseudo-real data through system-level testbeds such as those identified in [87]. As system-level testbeds are a smaller scale version of a real network, at least in theory, testbeds can execute almost all the necessary functions of a real network such as downlink, uplink, handovers, and scheduling to name a few. Testbeds offer the freedom of experimenting with different COP-KPI scenarios and can also provide open access to the data without privacy concerns. However, since testbeds are usually deployed on a small scale, the experiment range can be limited. Furthermore, particular design, architecture, and component specifications limit the type of experiments that can be conducted on testbeds to generate pseudo-real data. The generalization of models trained on testbeds data to a real network also remains an open research question. For these reasons, other techniques for data augmentation must be investigated.

- Synthetic data through simulators: A realistic system-level simulator could be

a viable option for generating synthetic data to augment the sparse real data. For instance, authors in [63] developed a system-level cellular network simulator named SyntheticNET and evaluated its features with existing system-level simulators such as ns-3 and Vienna. Utilizing this simulator, authors in [126] were able to generate synthetic data for a wide range of COPs and KPIs, through experimentation that would not have been possible to perform in a real network. To date, system-level simulators that possess all the traits of a cellular systems are scares. Therefore, additional complementing approaches for synthetic data generation are needed.

- Data augmentation through generative adversarial networks: Another promising approach for synthetic data generation is Generative Adversarial Networks (GANs) which has been extremely corroborated in the field of computer vision to create synthetic images [127]. In the context of cellular networks, studies utilizing GANs for tabular data augmentation remain scarce. One of the first studies that leveraged GANs for generating tabular cellular data is presented in [128]. GAN can generate very high-dimensional synthetic data for characterizing extremely complex systems such as cellular networks. However, one of the key challenges in exploiting GANs for addressing training data sparsity is the hyperparameter design challenge as previously discussed in Section 4.3.

On the other hand, in the absence of data enrichment techniques, utilization of the sparse data remains possible by leveraging models which are robust to training data sparsity.

Leverage models that are robust to data sparsity

- Transfer learning: GANs, matrix completion or other interpolation techniques identified above have limited utility if the latent features in the sparse data are too little. In these scenarios, the transfer learning (TL) paradigm can be

leveraged. In TL, the source domain and source learning task are leveraged to help improve the learning of the target predictive function. In the cellular network context, similarities among cells (i.e., traffic pattern and coverage area) can be leveraged to determine when to use TL. A key practical challenge here is to quantify the similarity to determine where to transfer the knowledge. Recently several studies have used transfer learning to address various aspects of cellular networks optimization [129].

- Federated learning: Federated learning is another model which can be leveraged for sparse training data. In federated learning, several clients collaboratively train a model without disclosing their data. The data from each client may be drawn from different distributions and can diverge the local learning process of the aggregate model, worsening the convergence rate and final performance of the aggregate model. A possible solution is the use of online reinforcement learning for federated hyperparameter selection. In this process, the learners perform an action by selecting hyperparameter values in each round and the learner gets a reward relative to the reduction in training loss at the end of each round. The hyperparameter selection policy is then updated to maximize the rewards.

4.4.2 Addressing the hyperparameter optimization challenge

Several promising hyperparameter optimization (HPO) methods can be used depending on the model being used, size, and type of the data.

- Grid and random search HPO: The grid search method explores the global search space by trying combinations of hyperparameters at regular predefined intervals. The size of the interval offers a tradeoff between the exhaustiveness of the search for globally optimal parameter combinations and the computational effort. On the other hand, the random search method randomly selects

possible hyperparameter combinations for a given number of iterations from a pool of hyperparameters. However, due to the exhaustive search nature of these approaches, they work best for models with few hyperparameters, such as k-nearest neighbor (KNN) which has 6 hyperparameters looking back in Fig. 4.2. In addition, they have a lengthy convergence period especially as the search space gets bigger.

- Bayesian model based HPO: Bayesian model-based HPO is a more efficient way of HPO especially for highly complex models such as tree-based ensemble models (i.e., Catboost) and deep neural networks (DNNs). Bayesian optimization minimizes the search space by focusing on a small region of hyperparameter search space where the most optimal combination is expected to lie. Although much efficient than grid and random search, Bayesian model-based HPO is not completely zero-touch as it still requires manual selection of the model (e.g., choosing between ML and DL models) and setting of the range of hyperparameters for creating a search space (e.g., setting the range of the learning rate of a DL model).

- Automated machine learning: Towards achieving the zero-touch HPO, automated machine learning (AutoML) is perhaps the closest method currently available and has been gaining a lot of attention due to its promising potentials. Although the majority of the current AutoML tools rely on Bayesian optimization, AutoML eliminates the need of manually choosing a model and setting a range of hyperparameters. For a given data, AutoML finds the best performing model and the corresponding hyperparameters without human intervention. Moreover, AutoML improves the reproducibility and fairness of scientific studies. It facilitates fair comparisons since different methods can only be compared fairly if they all receive the same level of tuning for the problem at hand. Nevertheless, since the model and range of hyperparame-

ters are not set in AutoML, most of the time, the search space is large and thus incurs a high computational cost.

- Neuro-evolutionary algorithm for deep learning design: Aside from hyperparameters, the structure of the architecture is an additional challenge for DL models such as DNNs. Neuro-evolutionary algorithms can be used to train and tune DNNs instead of the gradient-based process. They are biologically inspired mechanisms to improve iteratively upon a population of candidate solutions to the objective function. These algorithms can be used for learning the building block, hyperparameters, and architectures of the neural networks. In addition, they can be used to even determine the rules of learning. However, the DNNs constructed from neuro-evolutionary algorithms are often complex and unprincipled.

- Addressing the computational complexity of hyperparameter optimization: While the above-mentioned approaches can address the HPO challenges, they are usually computationally complex. This inefficiency can hinder the full realization of zero-touch optimization since the underlying models in the automation system need to quickly adapt to the changing network conditions. Moreover, the high utilization overhead and extreme memory utilization brought by the hyperparameter optimization turns both the equipment vendors and operators away from AI-based optimization solutions. Several methods to address the resource-intensive nature of HPO do exist such as one presented in [130], wherein the authors reduced the computational cost of neural AutoML by leveraging transfer learning. In the context of cellular networks, authors in [131] devised a coordinated hyperparameter search method that avoids the resource-hungry process of moving the training data from the edge to the central location, thus reducing the required time to mine the optimal hyperparameters.

4.5 AZTEK: A Framework for Zero-Touch Optimization in Emerging Cellular Networks

Using the outlined solutions above to address the limitations of manual network operation identified in Section 4.1, we propose a zero-touch system-level COP-KPI optimization framework for 6G network called AZTEK. Fig. 4.3 presents a schematic of the AZTEK framework. AZTEK is designed to have full intelligence of the current network status, the ability to predict user behavior, and the capability of dynamically associating the network response to the network COPs. More importantly, in contrast to most existing frameworks for zero-touch automation e.g. [123], this framework addresses the two key challenges (i.e., training data sparsity and model hyperparameter tuning) that hinder the full realization of the capabilities of AI in mobile networks.

AZTEK involves the following steps:

1. The first step for a data driven framework is usually data collection. AZTEK gathers the data for COP-KPI mapping from sources like minimization of drive test (MDT) reports, drive tests, and statistical gathering tools like Operations Support Systems (OSS) into an aggregate data set. Data from these sources will contain different network configuration parameters related to base station deployment (i.e., frequency band, height, propagation environment, etc.), antenna (i.e., tilt, azimuth, pattern, MIMO configuration, etc.), UE specific data (i.e., user location, measured signal strength and quality, etc.) with the corresponding system-level KPIs (i.e., throughput, utilization, handover success rate, etc.).

2. Since these data are usually sparse and not representative, we propose a data augmentation block that uses an assemble of techniques outlined in Section 4.4 to form a data lake. Simulator can generate synthetic data for different and wider range of COP-KPI combinations which won't be possible to gather in real network.

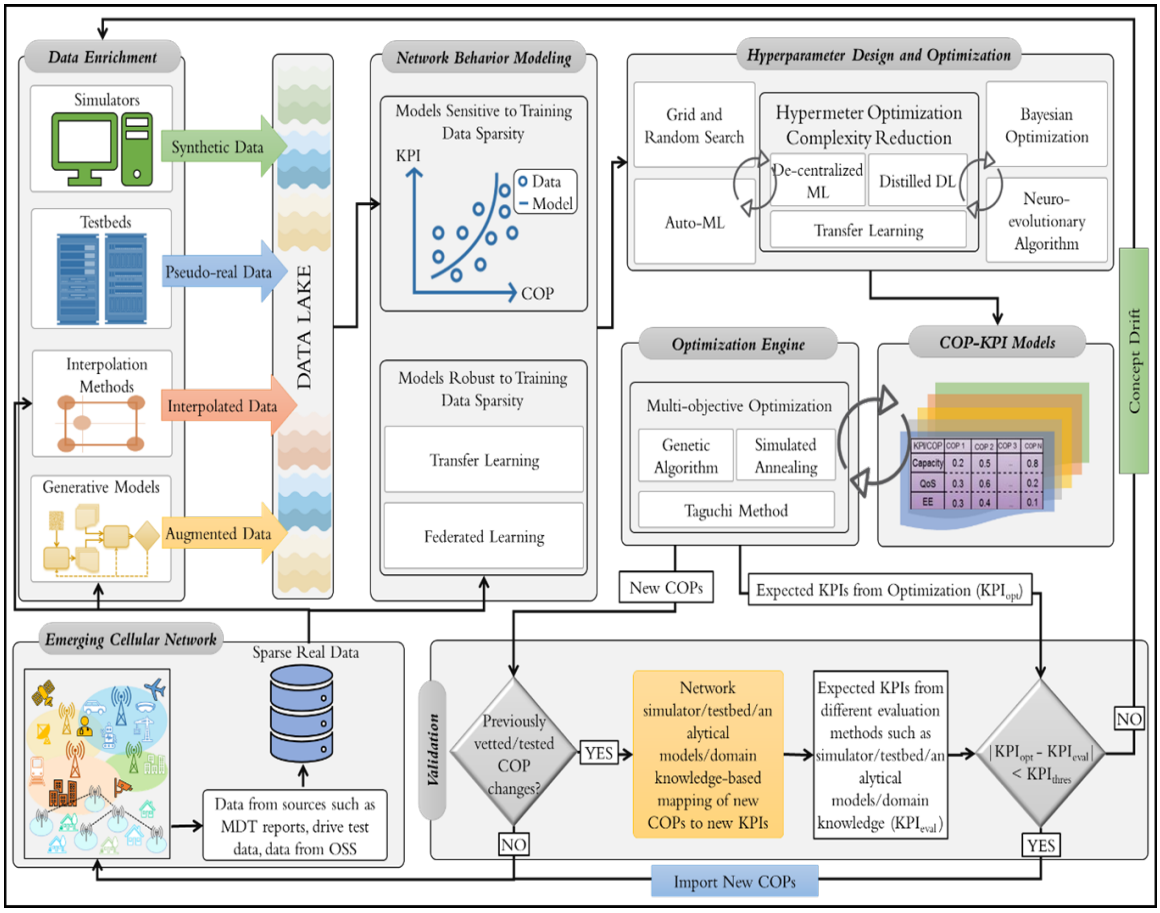


Fig. 4.3: AZTEK: Proposed zero-touch optimization framework addressing key challenges on the use of AI in mobile networks.

A more realistic set of COP-KPI combination can be generated using a testbed. On the other hand, generative models can be used to address the class imbalance in the data gathered (i.e., increase the number of anomalies such as hand over failure and radio link failure). Meanwhile, interpolation methods can generate missing values of measured signal strength and quality. This enriched data lake is a collection of pseudo-real, synthetic, interpolated, and augmented data to improve the representativeness before feeding them into the network behavior modeling block. Meanwhile, sparse data can also be utilized to directly train models that are robust to training data sparsity as outlined in Section 4.4.

3. ML, DL, or a combination of both forming ensemble models perform the modeling of the COP-KPI relationship. For instance, regression models can be used

to estimate the values of KPI given different set of COPs mainly in tabular form. Meanwhile, models such as convolutional neural networks (CNN) can be used for image-based COP-KPI relationship mining. The models which are sensitive to data sparsity are trained using the enriched data from the augmentation block while the models robust to data sparsity such as Federated Learning and Transfer Learning are directly trained using the sparse data.

4. To achieve optimal performance, hyperparameter optimization and complexity reduction techniques identified in Section 4.4 are leveraged.

5. A well-trained model, combined with an optimization engine, performs the COP-KPI optimization. Instead of relying on a hit-and-trial approach like the current SOTA SON, AZTEK relies on the model to determine the optimal set of COPs with the expected new KPIs. Multi-objective optimization using methods such as Genetic Algorithm, Taguchi Method, or Simulated Annealing is performed to estimate the new KPIs given the set of COPs.

6. New COPs and expected KPIs are passed into a validation block to determine the behavior of the network based on different evaluation methods such as simulations, testbeds, analytical modeling, or domain knowledge-based validation. The new COPs are only utilized if the evaluated behavior tallies the expected KPIs (i.e., the difference between the expected KPI from the optimization engine, KPI_{opt} and the expected KPI from the different evaluation methods, KPI_{eval} is less than a threshold, KPI_{thres}).

7. If validation in step 6 fails, there is feedback to the concept drift block, which in turn updates the model behavior. Even if validation returns a positive outcome, the concept drift block can be triggered periodically to maintain the accuracy of the model. The proposed framework keeps track of the past anomalies and can help identify any potential network failures.

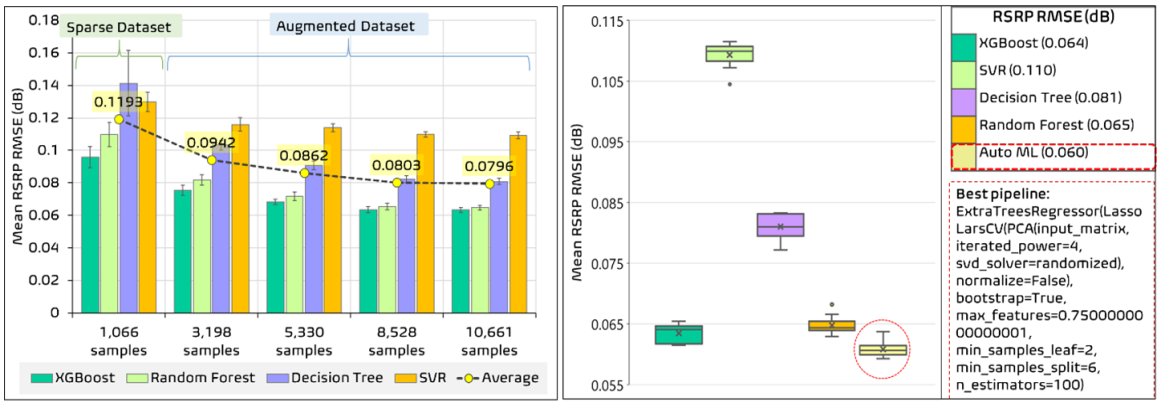
4.6 A Case Study on AI-based Zero-Touch Optimization in Emerging Cellular Networks

To demonstrate AZTEK framework, we analyze the optimization of the coverage KPI in the form of mean Reference Signal Received Power (RSRP) of all users in an observation area in this case study. We select several mobility COPs which affect this KPI including A5-Threhold1 (A5-T1), A5-Threshold2 (A5-T2), and A5 Time-to-trigger (A5-TTT). The selection of this COP-KPI set is done since no analytical models exist to provide insights into the nature of the relationship between the aforementioned COPs and the KPI of interest. The goal is to train a model that can capture the relationship between the COPs and KPI. The performance of the model is measured by the RMSE. We leverage the best model as input for an AI-based COP-KPI optimization solution which heuristically searches the optimal COP combination that maximizes the KPI.

We use SyntheticNET simulator [63] to generate an initial sparse data set of only 1,066 COP-KPI data samples. These data samples originate using industry gold standard ranges for A5-T1, A5-T2, and A5-TTT. Next, to address the data sparsity challenge through simulation-based data augmentation, we incrementally increase the amount of data by exploring wider ranges of COPs until finally, 4 richer sets of data with a varying number of samples are generated. The largest of which is a dataset consisting of 10,661 data points using a range of -90dBm to -120dBm for A5-T1 and A5-T2 with 5 values of A5-TTT (i.e., 64ms, 128ms, 256ms, 320ms, and 512ms).

We train and evaluate several classical AI models, namely XGBoost, Random Forest, Decision Tree, and Support Vector Regression, using the sparse and 4 sets of simulator-augmented training data. Performance evaluation results shown in Fig. 4.4(a) reveal that models trained with sparse data perform poorly, both individually and when averaged as a group, compared to AI models trained with augmented

data samples. Results also show that, in general, the performance of the AI models improves as the number of training data increases. The prediction error is reduced by 33% from an average of 0.1193dB when sparse data is utilized compared to 0.0796dB when enriched data is used in training the AI models.



(a) Addressing data sparsity challenge using simulators

(b) Leveraging Auto-ML to address the challenge of hyperparameter tuning

Fig. 4.4: The optimal performance of ML models is achieved by addressing the sparsity and hyperparameter optimization challenge.

Training the models with enriched data set only addresses the challenge of data sparsity. However, further exploitation of the full capabilities of these models requires addressing the hyperparameter optimization challenge. The capability of AutoML to automatically select the best model and hyperparameters with minimal human intervention makes it an ideal fit for the AZTEK framework. Thus, we leverage an AutoML tool called TPOT [132] and train it with the augmented data samples to come up with a model which outperforms other off-the-shelf AI models. Based on the hyperparameter optimization, the TPOT AutoML method generates an ExtraTreesRegressor model with optimal hyperparameters shown in Fig. 4.4(b). This model created by the AutoML exhibits the least RMSE of 0.060dB against other models with default hyperparameter settings.

The optimization block of AZTEK (see Fig. 4.3) leverages the results of the AutoML generated model to perform COP-KPI optimization using the Genetic Algorithm (GA). The role of GA is to efficiently search for the optimal COP combination that maximizes the KPI. The results in Fig. 4.5 show that GA converges quickly to

near-optimal values compared to brute force. GA only takes 350 iterations compared to 4805 iterations using brute force. The fast convergence time makes the solution agile especially for fast-changing network conditions and to meet the ultra-reliability and low latency requirements. Despite the guaranteed optimal value, using brute force is not computationally viable, particularly for large-scale problems, i.e., ones involving multiple parameters and base stations. The very small difference in the optimal KPI values between brute force (-109.30dBm) and GA (-109.41dBm) verifies that the presented solution, which combines AI for KPI prediction with heuristic optimization, enables a zero-touch self-optimization in 6G network.

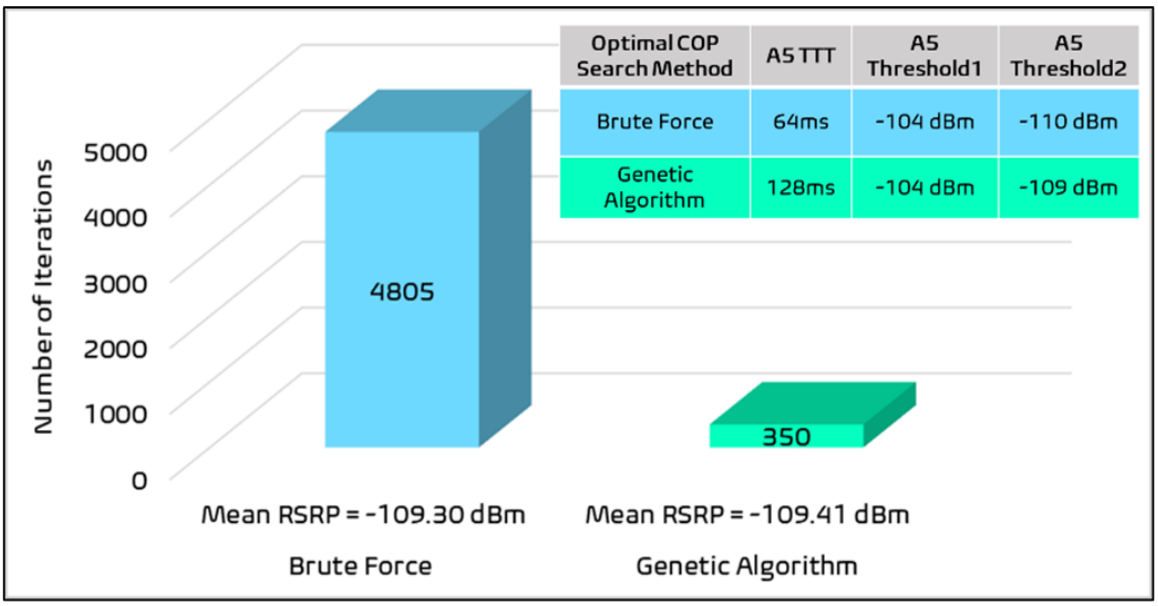


Fig. 4.5: Comparison in the performance between brute force approach and Genetic Algorithm.

4.7 Conclusion

We propose an AI-powered zero-touch optimization scheme for future 6G networks called AZTEK (AI-Enabled Zero-Touch System-Level KPI Optimization). AZTEK aims to eliminate the need for manual, hit-and-trial-based KPI optimization. The realization of proposed zero-touch automation hinges on addressing two key challenges which undermine the capability of AI. 1) Sparsity of training data particularly

in the mobile network industry and 2) the hyperparameter design and optimization of the AI models. For the first challenge of data sparsity, we identify several training data enrichment solutions including interpolation methods, pseudo-real data generation, and synthetic data generation. In addition, we explicate the strengths as well as the limitations of each technique. For the second key challenge i.e., AI-model hyperparameter optimization, we identify a set of solutions grouped based on their applications. Among these solutions, AutoML is currently the most promising tool that can facilitate a zero-touch optimization process. We demonstrate the viability of the proposed zero-touch automation framework through a case study. First, we leverage the power of custom system-level simulators to generate a wide range of COP-KPI data to address the data sparsity challenge, and second, we develop a model with optimal hyperparameters using AutoML. The use case results show the potential of AI to enable a zero-touch optimization process in future 6G networks.

CHAPTER 5

An ML-based Framework Towards Improving the Mobility Robustness in Emerging Cellular Networks

5.1 Introduction

In Chapters 2, 3 and 4 we present several platforms and solutions to enable a zero touch network optimization. However other challenges still exists that impacts the mobility performance of emerging cellular networks such as radio link failures (RLF). RLFs caused by handover failures are becoming more pressing in emerging cellular networks with network densification, multi-band operation, and the rising complexity of handover parameter optimization. One way to reduce the RLF and to improve HOSR is to manually tune and optimize HO related parameters, i.e., hysteresis, cell individual offset (CIO) and, time-to-trigger (TTT) through hit and trial. Additionally, another industrial practice is the tuning of parameters such as tilt, azimuth, and transmit (Tx) power, which helps alleviate the RLF issues by improving the coverage and reducing interference [133]. However, such manual and mostly hit and trial-based parameter tuning is very time-consuming as the statistics have to be observed in live network for hours or days to see if the new parameter value is better than earlier. Also, dependency on human intervention and experience based incomplete or heuristic understanding of the system behavior, makes this approach prone to errors. Some self-organizing network (SON) functions such as mobility robustness optimization (MRO) can automate this process to some extent by automatically adjusting the CIO values based on prior HO performance [134]. Most state-of-the-art MRO solutions effectively automate the hit and trial-based tuning as a closed loop eliminating the need for manual labor. However, the fact

that these SON solutions do not leverage any in-depth modeling and analysis of the system behavior, means their performance is not guaranteed to be any superior than manual hit and trial-based parameter tuning.

5.1.1 Relevant Work

Several studies have been done on handover failure prediction. Authors in [135] proposed a neural network capable of predicting the future Reference Signal Received Power (RSRP) values of the source and target BSs operating in similar frequency. They cascaded the RSRP prediction model with another neural network which acts as a classifier to determine if the handover will fail or not. Meanwhile, authors in [136] devised a new handover algorithm built from RSRP prediction of two remote radio units using an improved grey verhulst model. The authors proposed to do the handover decision at early step based on the predicted RSRP values. A study on forecasting future handovers is performed in [137] based on fuzzy forecasting model where authors used historical RSRP data to forecast future RSRP. While results are promising, the effectiveness of [135–137] heavily relies on RSRP prediction accuracy using historical data. However, signal condition such as RSRP experiences frequent fluctuations making its behavior difficult to predict and capture [138]. This intrinsic randomness in RSRP values makes RSRP prediction-based approach in predicting HOF precarious.

Literature on improving handover performance through HOF mitigation can be broadly grouped into two themes. The first theme involves utilizing the current HO standard procedure while optimizing HO-related parameters [1, 15–19]. The second theme involves proposal of completely new HO algorithms [20–23].

Authors in [15] presented a fuzzy logic controller that modifies HO margin (HOM) to optimize call dropping ratio caused by HOFs. Meanwhile, optimization of HO parameters such as TTT and CIO to improve HOF and ping-pong has been pro-

posed in [16]. The authors clustered users depending on the RSRP trend when users move from indoor to outdoor location. Authors in [17] used reinforcement learning to optimize parameters such as handover threshold, CIO, hysteresis, and TTT. Almost similar sets of parameters (HOM, CIO, and TTT) are analyzed and optimized in [18] with the addition of inter-site distance. Meanwhile, a single parameter, HOM, is optimized in [19] to improve HOSR taking into account the speed of the users. Authors in [1] tuned CIO for each problematic cell-pair and showed improvement in handover success rates on a real network setting. Although the results are encouraging, one caveat in HOSR improvement via optimizing HO-related parameters is that it only resolves HOFs caused by untimely handovers (i.e., too late or too early handovers). This approach fails to take into account another, and rather the main reason for HOFs and thus poor HOSR i.e., the poor signal quality and undesirable propagation condition [139].

A new HO scheme is presented [20] using a two-level neural network which predicts how user experience is affected when HO is performed to a particular BS. In [21], authors proposed a pre-handover algorithm aided by mobility prediction using Gauss-Markov mobility model to improve HOSR in Long Term Evolution (LTE). Authors in [22] proposed a new HO scheme that minimizes handover failures by reducing the HO interruption time. Another new HO algorithm is devised in [23] to improve the handover performance of users in femtocells. Authors in [23] used a completely different set of parameters compared to currently standardized set of parameters involved in HO. These include received signal strength, user speed, traffic type, and bandwidth. Although the results are promising, the proposals made in [20–23] are complex and require a change in the current HO standards.

Studies proposing a complete solution to mitigate HOF by leveraging a HOF prediction model are scarce. The idea is first mentioned in [135]. However, in contrast to our proposed solution that builds a HOF prediction model and uses it for enabling

a HOF mitigation scheme, the study in [135] only presents a HOF prediction model and does not present a scheme to mitigate HOFs. Meanwhile, closest to our presented solutions is the work on optimization of MRO SON function in [140] where prediction of HOF is studied for improving the HOSR. However, proposed scheme uses a simple threshold-based approach for predicting HOF. It does not leverage the power of AI to predict HOF with potentially better accuracy.

Lastly, most of the current handover prediction and mitigation approaches in the existing literature are focused on intra-frequency handovers. However, in contrast to intra-frequency where both the source and target BS are on one frequency band, in inter-frequency source and target base stations have different frequencies. In inter-frequency HO the user can measure the RSRP from the target cell without switching to new frequency. For inter-frequency HO user has to periodically scan (at preset intervals) for other frequencies to determine potential target cells for HO. The involvement of higher number of frequency layers in inter-frequency HO necessitates the use of more input parameters into the HOF prediction model compared to intra-frequency HOF prediction model. This makes the current intra-frequency HOF prediction models ineffective for the inter-frequency HOF prediction. Additionally, to the best of author's knowledge, there is no study in the literature that aims at mitigating inter-frequency HOF. The importance of analyzing and improving inter-frequency HO further increases with the advent of 5G&B that brings with it multiplicity of frequency layers.

5.1.2 Proposed Approach and Contributions

To address the limitations of the current solutions in literature, we present a first of its kind framework, named TORIS (Transmit Power Tuning-based HOSR Improvement Scheme). TORIS consist of two components, an AI-based HOF prediction module and Tx power tuning scheme, that work in tandem to substantially reduce

the HOF rate, compared to state-of-the-art schemes. Unlike other methods which rely on historical RSRP data, TORIS's AI-based HOF prediction module leverages measurements which are actually reported by the user to predict potential HOF occurences. By doing this, TORIS eliminates the potential error from frequent fluctuations of radio signal condition. The advantages of the proposed Tx power tuning scheme stem from its two key attributes: 1) Compared to the state-of-the-art HOF mitigation schemes that tweak CIO, the proposed scheme addresses one root cause of HOF i.e., poor signal quality by tuning Tx powers instead of CIOs. 2) Proposed scheme is simple and does not require any change in standard and thus can be easily implemented in real 4G, 5G and beyond networks.

TORIS framework is illustrated in Fig. 6.4. TORIS starts with data generation and collection from sources like minimization of drive test (MDT), drive test and Operations Support Systems (OSS) data as depicted in the top left block in Fig. 6.4. In this chapter, data generation is done using a 3GPP-compliant system level simulator. As in a typical commercial real network, HOF events happen far less frequently compared to handover success events, the training data that can be harnessed to train a HOF prediction model is bound to be extremely imbalanced. This imbalanced data, if used as it is for training an AI model, can lead to a model biased towards misclassifying all or most HO events as HO success events. To address this challenge, TORIS leverages several data augmentation and synthetic data generation techniques as depicted by the TORIS modules in 2nd row and middle column of Fig. 6.4. These techniques include Synthetic Minority Oversampling TEchnique (SMOTE), Chow-Liu Bayesian Networks (CLBN), and Generative Adversarial Network (GAN) based deep neural networks combined with border line sampling concept. However, the data generated using these techniques are not readily usable. By incorporating a data evaluation process (top right block in Fig. 6.4), TORIS ensures that the quality of the generated data is acceptable. Synthetic data with good quality are used to train different ML models providing a multi-faceted performance

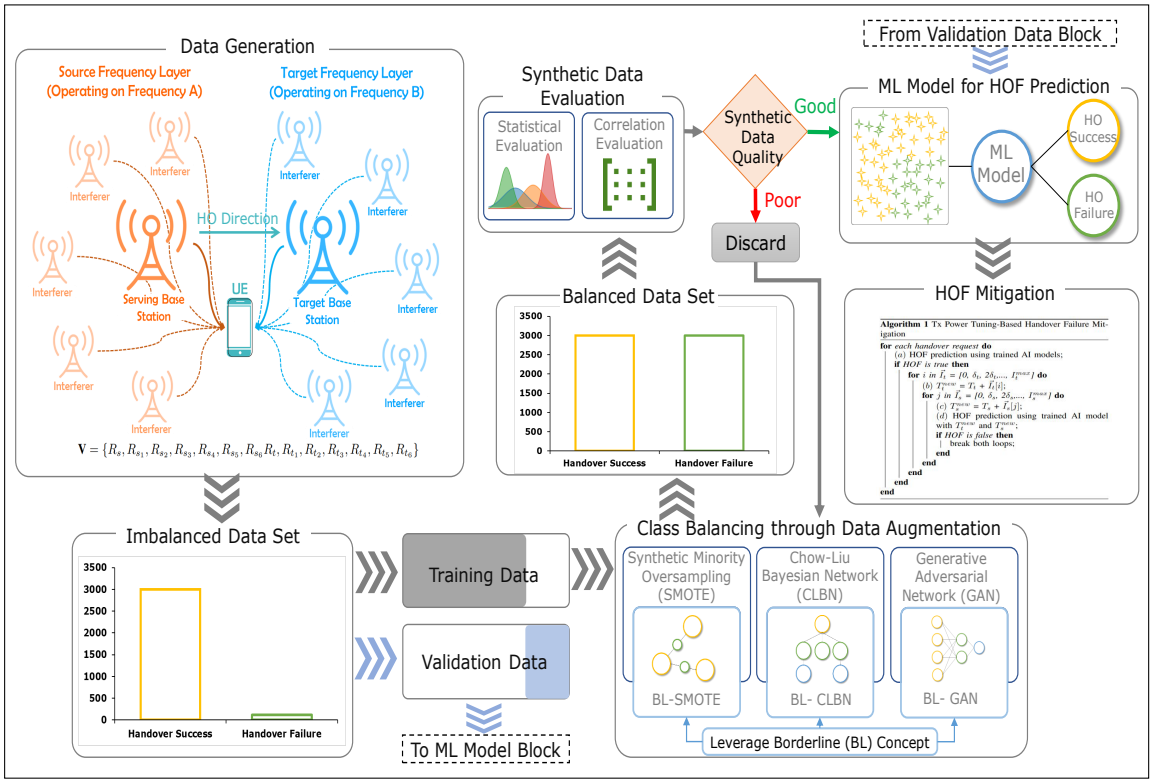


Fig. 5.1: TORIS (Transmit Power Tuning-based HOSR Improvement Scheme) framework.

comparison of Naive Bayes Classifier, K-nearest Neighbors (KNN), Support Vector Classifier, Random Forest, Extreme Gradient Boosting (XGBoost), and Adaptive Boosting (AdaBoost) for predicting HOFs. These state-of-the-art methods are selected based on their effectiveness as binary classifiers and their ability to work well specially on small data sets compared to deep neural networks, in addition to the faster training which is crucial for HOF prediction. Further analysis of the advantages of the selected six techniques for binary classification, compared to other ML methods, reader is referred to [141–143]. Whenever ML detects potential HOF, the HOF mitigation scheme kicks in (3rd block from top right) and rectify the failure by heuristically varying the transmit power of the involved BSs.

The main contributions of this chapter can be summarized as follows:

1. This chapter presents a novel framework, named TORIS (Transmit Power Tuning-based HOSR Improvement Scheme). To the best of the authors knowl-

edge, this is the first study to present such solution as current studies analyze HOF prediction and HOF mitigation in silos. To improve the accuracy of the TORIS's HOF prediction model we identify a novel set of features by leveraging domain knowledge and insights gained from Sobol sensitivity analysis of the raw data. This novel feature set includes features like signal strength in the form of RSRP, signal quality in the form of geometric factor (G-Factor) and user speed. These smart features improve the performance of the HOF prediction model compared to the existing models that rely only on the RSRP of the serving and target BSs [135].

2. A major challenge in HOF prediction model design is the imbalance in the real network-based training data where the number of HOF instances are far less compared to HOS instances in the gathered data. We investigate an extensive set of data augmentation techniques to address the class imbalance problem by preprocessing and augmenting the training data. We exploit an unconventional AI-based data generation technique in the form of GAN. To the best of the authors knowledge, this is the first attempt to leverage GAN in addressing the class imbalance challenge in context of cellular network. We analyze the quality of the augmented data generated by the data balancing techniques using diverse metrics. These metrics reveal that, despite their popularity, the inherent dependency of deep learning-based methods, such as GAN, on large training data make them unsuitable to generate synthetic minority data for this particular case. To further improve the performance, we leverage the borderline concept for addressing the class imbalance challenge.

3. This chapter investigates the potential of select AI techniques for creating HOF model that provide trade-off between complexity and accuracy using the augmented as well as raw data. Results show that AdaBoost yields the best performance and thus should be a candidate of choice for creating the

HOF prediction model component of TORIS in real networks.

4. This chapter presents a heuristic scheme for mitigating HOFs. This scheme constitutes the second component of TORIS and uses prediction of HOF failure model to dynamically adjust the Tx power of serving and target BSs. Results based on 3GPP compliant simulator show that the proposed HOF mitigation scheme can reduce HOF by 75%. It also leads to substantially better performance compared to the state-of-the-art CIO based schemes in literature [1].

The rest of the chapter is organized as follows: Section 5.2 discusses handover success rate including the factors affecting the handover performance, simulation setup and data generation. In Section 5.3, we present the data augmentation techniques to address the class imbalance issue as well as the evaluation metrics to check the synthetic data quality. Section 5.4 presents the HOF prediction models and the domain knowledge-based feature engineering approach. Meanwhile, Section 5.5 presents the HOF mitigation algorithm and its evaluation followed by the results and discussion in Section 5.6. Finally, Section 5.7 concludes this chapter.

5.2 System Model and Data Generation

This section describes inter-frequency HOSR and the factors affecting the handover performance. It is followed by the presentation of the end-to-end framework for improving the HOSR by leveraging HOF prediction before discussing the simulation setup and data generation process.

5.2.1 Inter-frequency Handover Success Rate

In current 4G and 5G networks, there are two kinds of handovers namely intra-frequency handover and inter-frequency handover. The former happens when a

user switches from one BS to another operating on the same frequency band. The latter happens between the cells operating on different frequencies. Inter-frequency handovers also happen when the UE requests a service which is not available at current frequency or during load balancing across the frequency bands. Inter-frequency HO utility is expected to expand due to the co-existence of 4G and 5G at different bands. This signifies the importance of inter-frequency HOSR optimization for the current and future cellular networks. However, despite its growing significance and associated open challenges, inter-frequency HOSR enhancement remains an under investigated topic as most literature remains focused on intra-frequency handover.

Inter-frequency HOSR is a direct measure of the handover performance between different frequency bands. For this reason, HOSR is one of the major KPI of interest in the current and emerging cellular networks. High HOSR means that the movement of users from one BS to another is seamless and thus, satisfactory QoE is maintained. In addition, URLLC, which is one of the use cases of 5G and beyond heavily relies on seamless handovers particularly for applications such as intelligent transport systems and autonomous cars. HOSR ξ can be described as:

$$\xi = \frac{n_s}{n_s + n_f} \times 100\%, \tag{5.1}$$

where n_s and n_f are the number of successful and failed inter-frequency handovers, respectively, in the network.

5.2.2 Factors Affecting Handover Success Rate

Aside from sub-optimally configured handover parameters [1, 15–19], one of the major factors affecting HOSR is signal strength and quality of both the source and the target BSs. As previously mentioned, handovers are executed with the help of interchanging commands and acknowledgments between the UE and the BS. During this crucial period, the connection between the UE and the BS should be

good enough to maintain the back-and-forth exchange of HO-related messages. In current cellular network, signal strength is measured by the parameter called RSRP. The downlink RSRP R_u^x for a user u connected to the serving BS trying to perform HO to the target BS is given by:

$$R_u^x = T_x d_u^x, \tag{5.2}$$

where x corresponds to either serving or target BSs, T is the transmit power of the BS and d is the path loss dependent component of the user u. The path loss dependent component also contains antenna gains as well as the shadowing for the user, which is modeled as a Gaussian random variable.

Meanwhile, the quality of the signal in a cellular network can be measured using several metrics. In this chapter,we use signal quality metric known as G-Factor. G-Factor can be defined as the ratio between the RSRP and the combined signal power of interfering cells in downlink. Mathematically, G-Factor is expressed as:

$$G_u^x = \frac{R_u^x}{\sum\limits_{\forall i \in I_x} R_u^i + n_0}, \tag{5.3}$$

where G is the G-Factor, x is either the serving or target BS, the set I_x contains 5 strongest interferers for the BS x and n_0 is the noise floor constant. The selection of 5 strongest interferers is based on the industrial domain knowledge of the authors. Based on our field experience of working with real 4G and 5G networks, the interference from 5 neighboring base stations contribute the most in the calculation of G-Factor and the impact diminishes from sixth interferer on wards.

Although signal-to-interference-plus-noise ratio (SINR) is also a good metric to measure the signal quality, it is sensitive to instantaneously changing load of the neighboring BSs. To eliminate this dependency, we use G-Factor instead. In addition, as G-Factor is basically the worst case SINR, using it as a metric to predict the HOF will cover worst case instances which might be missed if SINR is used as

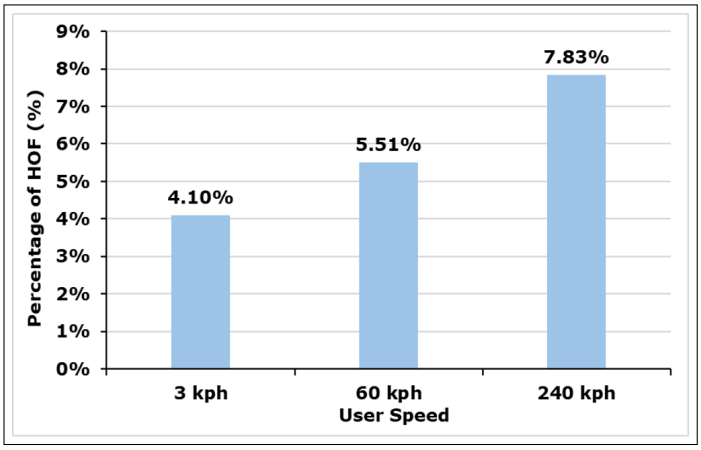


Fig. 5.2: Impact of user speed in occurrences of HOFs.

signal quality metric.

In addition to the signal strength and quality, we have also used user speed as an input feature for HOF prediction. Speed is shown to have an impact on handover performance [144]. This study showed a decrease in handover success rate with the increase in user speed. Our own analysis on the effect of user speed confirms the finding in [144]. Fig. 2 shows the increase in the percentage of HOFs with increase in the speed of the users.

5.2.3 Simulation Setup and Data Generation

Collecting the necessary data from a live network though plausible in theory, is impractical in practice. Due to the sporadic nature of HOF occurrence, numerous drive tests are necessary to gather enough samples. This process of gathering data samples is an extremely resource-intensive task. In this backdrop, to generate the data, we exploit SyntheticNET [63]. This is the first simulator to model 5G mobility parameters in detail as needed for this study. SyntheticNET simulator has been calibrated against real network measurements to ensure the validity of the data generated through it.

An area of size 5km x 5km is used for the simulation as shown in Fig. 5.3. A multi-carrier heterogenous network composed of two frequency layers is deployed inside

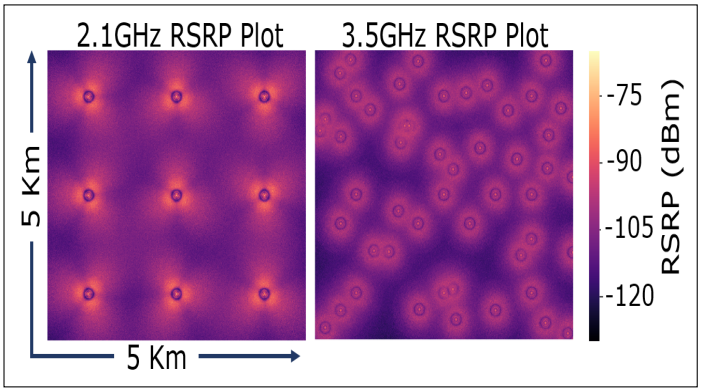


Fig. 5.3: Network deployment layout and RSRP plot for simulations.

Table 5.1: Description of Simulation Parameters

Parameter Description	**Value**
Simulation area	25 km^2
Number of 2.1GHz macro transmitters	27
Number of 3.5GHz small cells	50
Macro cell height	30 m
Small cell height	20 m
Macro cell Tx power	40 dBm
Small cell Tx power	30 dBm
Pathloss exponent	3
Shadowing standard deviation	4
Active user density λ_u	12 per km^2
Speed vector S	[3, 60, 240] km/h
Transmission time interval (TTI)	1 ms
Total simulation time	15000 ms

the area. Macrocells operating at 2.1GHz, with 3 sectors each, are deployed using grid pattern. Meanwhile, small cells operating at 3.5GHz, with omni-directional antenna, are deployed in the network following a uniform random distribution. The initial deployment of the users in the network follows a uniform distribution with user density λ_u. Each user can move in the network with speed s_u chosen from a set S. All elements of the set S are equally probable, and the speed value remains constant for a user. The user mobility type is a random way point. The network level simulation parameters are summarized in Table 6.1.

The initial mobility parameters are chosen based on the GS setting of one of the leading operators in the USA. Event A2, which is triggered when RSRP of the

serving BS becomes lower than a threshold, is used to trigger the measurement gap as a pre-requisite for inter-frequency handover. In the simulations, we used -100 dBm for A2-threshold. Meanwhile, to trigger the HO process, event A3 is used. This event is triggered when the RSRP of the target BS becomes better than the serving BS's RSRP by some offset value. A3-offset of 2dB is used. TTT of 16ms and hysteresis of 0dB was used for events A2 and A3.

To gather data for handover failure prediction, we run the simulation for around 15000 TTIs. While moving, the users inside the simulation area measure and log the RSRP condition of the serving as well as the neighboring base stations in both frequency bands in the form of a state vector ($\mathbf{V}$) which can be expressed as:

$$\mathbf{V} = \{R_s, R_{s,n1}, R_{s,n2}, R_{s,n3}, R_{s,n4}, R_{s,n5}, R_t, R_{t,n1}, R_{t,n2}, R_{t,n3}, R_{t,n4}, R_{t,n5}\}, \quad (5.4)$$

where R is the reported RSRP, subscripts s and t correspond to the serving and target BSs operating on different frequency bands, respectively and $n_i \forall\ i \in 1, 2, 3, 4, 5$ relate to the interfering neighbors in the serving and target frequency layer sorted in descending order. When handover condition is triggered, the base station starts the HO process. After the HO execution, the measured RSRPs are flagged with either handover success or handover failure depending on the outcome of the handover. The resulting combination of RSRPs and handover status (i.e., failed or successful) are used to train HOF prediction models (after going through the data augmentation process to address the class imbalance issue). After the HOF prediction model is trained offline, it is imported into the handover failure algorithm to make online inference. A new simulation with similar setting as the simulation done for HOF prediction data generation is run with the addition of the HO failure mitigation algorithm and handover performance is observed. It is worth highlighting that similar simulation setup is used for objective evaluation of the gain of the proposed scheme. The developed HOF prediction model is expected to work

in varying deployment scenarios as it relies on user speed, RSRP, and G-factor of both the serving and target BS, and not on the actual deployment scenario or other configuration parameters in the simulation setup.

5.3 Data Augmentation and Evaluation of Synthetic Data Quality

In this section, we analyze the class imbalance issue and solve this problem using several synthetic data augmentation techniques. As a form of sanity check for the generated synthetic data to address the class imbalance issue, we present methods of evaluating the synthetic data quality.

5.3.1 Effect of Imbalance Data in HOF Prediction

Although handover failures are common, the occurrences are far lesser than the number of successful handovers. Since the simulator we used models a realistic HO scenario, our data set follows a similar trend. Majority of the samples are HOSs with very few HOFs. A classification model trained on the imbalanced data set might lead to the model poorly learning the decision boundary. The classification model becomes biased towards the majority class, making the prediction in favor of the majority class (i.e., HOS) while neglecting the more important minority class (i.e., HOF). Fig. 5.4 shows the impact of the imbalanced data set on the accuracy of some state-of-the-art classifiers. Worst classifier, Support Vector Classifier in this case, fails to predict even a single instance of HOF correctly. Meanwhile, two best classifiers Random Forest and AdaBoost are able to correctly classify only 21/36 (58.3%) and 20/36 (55.5%) HOFs, respectively.

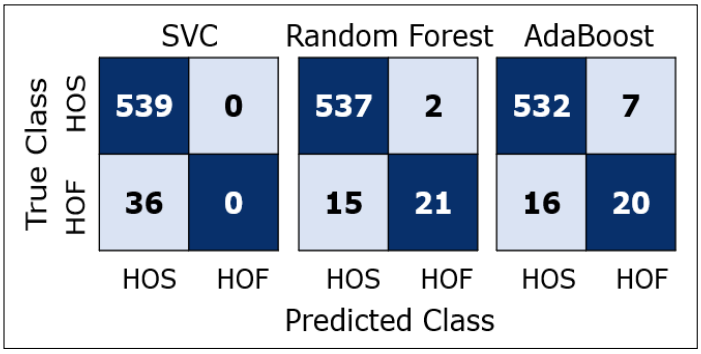


Fig. 5.4: Confusion matrix of the classifiers trained using imbalanced data set.

5.3.2 Data Augmentation Techniques to Address Class Imbalance

The class imbalance issue in classification can be solved by oversampling the minority class. Generating more samples of the minority class can improve the decision boundaries by making them less specific to the majority class. This wider decision boundaries improve the classifier generalization of the data set which ultimately improves the classification performance. In this chapter, we exploit three types of synthetic data generation techniques. First, we leverage the most commonly utilized technique in synthetic data generation for imbalanced data set called SMOTE, originally proposed in [145]. We use SMOTE for its well-known ease of implementation and interpretability. Second, Bayesian Network (BN) is another type of synthetic data generation technique, which creates an acyclic graph using probabilistic models. We choose BN for this study as it is simple, fast, and known to work efficiently even with small data samples [146]. We use Chow-Liu [147] method that approximates the tree with first-order dependency, which has the smallest Kullback–Leibler (KL) divergence to the actual joint probability distribution. In this chapter, we utilized the algorithm given in [148] for CLBN-based data generation. Finally, we use some newer synthetic data generation methods in the form of GAN. We utilize GAN due to its recent promising results in augmenting tabular numerical data. In this chapter, we leverage conditional tabular GAN (CTGAN) recently proposed in [149] to generate synthetic data. This particular type of GAN has been shown to outperform other variation of GANs for tabular data.

Table 5.2: Hyperparameters Used for Training CTGAN

Parameter Name	Values Evaluated	Optimal Setting
(Embedding Dimension) - (G Dimension) - (D Dimension)	(128) - (256,256) - (256,256) (64) - (128, 128) - (128,128) (32) - (64, 64) - (64, 64) (16) - (32, 32) - (32, 32) (8) - (16, 16) - (16, 16) (4) - (8, 8) - (8, 8)	(32) - (64, 64) - (64, 64)
Adam Optimizer Weight Decay	1e-01, 1e-03, 1e-05, 1e-07	1e-07
Batch Size	10, 30, 50, 70	50
Epoch	100, 300, 500, 700	500

For optimal performance, we tune and evaluate several hyperparameter combinations of CTGAN as shown in Table 5.2. Based on this list of hyperparameters, we perform grid search to find the best hyperparameter combination. We perform the evaluation by generating synthetic data using the different settings of CTGAN and training the AI-model to predict HO failures. We select the hyperparameter combination that produces the highest area under receiver operating characteristic (AUROC) curve and area under precision recall curve (AUPRC). The best performing CTGAN hyperparameters shown in Table 5.2 are selected and the performance in predicting HO failures is compared with other data augmentation techniques.

5.3.3 Data Augmentation Leveraging the Borderline Concept

Since its debut, several variations of SMOTE have been devised to improve its performance. One such variation is called Borderline-SMOTE (BL-SMOTE) introduced in [150]. BL-SMOTE has been proven to improve the performance of standalone SMOTE based on studies conducted by authors in [151, 152]. BL-SMOTE works on the principle of SMOTE with one major modification. It uses KNN to identify the samples near the decision boundary and over sample them instead of blindly oversampling all the minority data points. This improves the performance of SMOTE by limiting its action on samples which actually require oversampling.

Inspired by this concept, we implement the borderline technique with CLBN and GAN which we call BL-CLBN and BL-GAN, respectively. Just like BL-SMOTE, an additional step is implemented that first determines the minority samples near the class boundary using KNN. These samples are then fed into the CLBN and GAN to generate synthetic data. This method provides more resolution on the important area to facilitate better performance than simple CLBN and GAN.

5.3.4 Evaluation Methods for Synthetic Data

A thorough evaluation of the synthetic data generated from the aforementioned techniques is required before using it for model training because synthetic data, in some cases, are not accurate representation of the actual data. To ensure the synthetic data represent the actual data distribution, we conduct two types of evaluations to examine the quality of the generated data described below.

Statistical Evaluation

Earth Movers Distance (EMD) also known as Wasserstein Metric is used to measure the similarity between the distribution of real and synthetic data. The lower the EMD is, the closer one distribution is to the other. EMD is considered more robust compared to other distance metrics such as KL divergence and Jensen–Shannon divergence especially when the two distributions are disjoint or not in the same probability space [153]. Mathematically, weight transferred EMD (ζ) can be expressed as:

$$\zeta = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{W}_{ij} d_{ij}, \tag{5.5}$$

where $\mathbf{W}$ is a matrix and each element corresponds to the amount of mass transferred from one distribution in the ith location to the other distribution in the jth location while d is a distance affinity matrix which corresponds to the cost of moving a single element of mass from the ith location of one distribution towards the

jth location of the other distribution. This distance is calculated as the Euclidean ground distance between the two distributions.

Pairwise Correlation Comparison

Although EMD can capture the deviation of the distributions between real and synthetic data, it cannot capture the changes in the relationship between each input variable. To address this problem, we use pairwise correlation difference (PCD) as an additional evaluation method of synthetic data. The PCD metric measures the difference in the Frobennius norm ($||\ ||_F$) of the Pearson correlation ($Corr$) between real and synthetic data sets [154]. Mathematically, PCD is defined as:

$$PCD(M_{real}, M_{synt}) = \| \ Corr(\mathbf{M}_{real}) - Corr(\mathbf{M}_{synt}) \ \|_F, \tag{5.6}$$

where $\mathbf{M}_{real}$ and $\mathbf{M}_{synt}$ are the real and synthetic data matrices, respectively. Good synthetic data should have close to zero entries inside PCD matrix. Additionally, to represent PCD using a single variable instead of a matrix, we use the variable μ which corresponds to the mean absolute value of the elements in the PCD matrix. Mathematically, μ is defined as:

$$\mu = \frac{1}{n} \sum_{c=1}^{C} \sum_{r=1}^{R} |PCD(M_{real}, M_{synt})_{rc}|, \tag{5.7}$$

where $|PCD(M_{real}, M_{synt})_{rc}|$ is the absolute value of an element from the PCD matrix in rows r and columns c while R and C are the number of rows and columns of the PCD matrix, respectively. Meanwhile, n is the total number of matrix elements.

5.4 AI-Based Handover Failure Prediction Model

In this section, we discuss the HOF prediction approach and the AI models we leverage in this chapter. We also present the domain knowledge aware feature

engineering to improve the prediction performance against state-of-the-art solution.

5.4.1 HOF Prediction Models

We model HOF prediction as a binary classification problem. With classification, labels of the data points are predicted by mapping input features $\mathbf{V}_o$ to discrete labels y. In our study, input features $\mathbf{V}_o$ presented in eq. (5.8) include the measurements that a UE sends to the BS which trigger handover. These include the RSRP values of the serving and target BS as well as up to 5 strongest interferers for both serving and target layer. UE can report these RSRP values when the event is triggered. The RSRP values are saved at the start of the TTT when the entering condition of the event becomes true for the first time. Each of the input combinations $\mathbf{V}_o$ are labeled as either HOS or HOF. This HOS and HOF represents the discrete class y. Since the simulator is already calibrated against real network, the general insights drawn from the model are expected to hold for a real network. Nevertheless, for practical use of this solution, new model will have to be trained for each deployment scenario using real data where possible.

In this chapter, we evaluate the performance of six classification techniques namely Support Vector Classifier, Naïve Bayes, KNN, Random Forest, XGBoost, and AdaBoost. These state-of-the-art methods are selected based on their effectiveness as binary classifiers and their ability to work well specially on small data sets compared to deep neural networks, in addition to the faster training which is crucial for HOF prediction. Below are the descriptions of each of the ML model leveraged:

- Support Vector Classifier (SVC): SVC is a type of supervised ML technique which is often used for binary classification problems. Basically, SVC uses a hyperplane which is designed to classify training vectors into two classes. Its popularity can be attributed to its simplicity, speed and effectiveness to work

very well with limited amount of data points. However, SVM works best on data sets which are linearly separable.

- Naive Bayes Classifier (NBC): As the name implies, NBC works on the basis of Bayes theorem.This ML algorithm assumes independence among the input features. That is, each feature in a class in unrelated to other features making each of them contribute independently to the classification action. NBC are simple, easy to build and works best for large data sets.

- k-Nearest Neighbors (KNN): One of the simplest ML algorithm, KNN uses the concept of similarity measure such as distance function to make classification decisions. It stores all the available cases and makes classification based on how the neighbors of new data point are classified.

- Random Forest (RF): Random Forest is another kind of tree-based learning algorithm. RF classifier is composed of sets of decision trees. These trees are built using randomly sampled subset of the whole training data. The final class is decided by majority voting from all the class results from each decision tree. RF is considered as one of the most reliable algorithm for classification problems producing high accuracy even with small data set and large input features.

- Extreme Gradient Boosting (XGBoost): XGBoost is a popular type of gradient boosting algorithm which belongs to a ML category known as ensemble learning. Techniques under this category train several learners to perform the same task. In XGBoost, multiple regression trees, called weak learners, are trained and then converted into a single superior learner, which is the combination of the decision results of all the weak learners.

- Adaptive Boosting (AdaBoost): The weak learners in AdaBoost are decision trees with a single split, called decision stumps. AdaBoost works by putting

more weight on difficult to classify instances and less on those already handled well.

5.4.2 Domain Knowledge Based Feature Engineering

The performance of an AI model depends on the quality of the input features. Combined with domain aware smart selection of the input features, quality of the AI model can be improved. To this end, we begin by performing a detailed exploratory analysis to extract the features that influence the handover failures the most. As a first step, we perform the Sobol-based variance sensitivity analysis [155]. The findings of which is presented in Fig. 5.5. It is observed that the RSRP of the source BS has the highest impact on the performance of the prediction model followed by the RSRP of the target BS while the Top 2 interferers for the source and target are in third and fourth spots, respectively. Using these insights from the Sobol analysis, we identify sets of input features, which can potentially predict the HOF. The first feature set consists of the most important feature (source RSRP), while the second set consists of the two most important features (source and target RSRP). The feature set 2 has already been used to predict HOF in a previous study [135]. Set 4 consists of four most important features (source RSRP, target RSRP, top interferers of the source and target BS). Feature set 6 contains all the 12 input features without any intelligent feature selection. We leverage domain knowledge to incorporate the impact of all interferers using a single parameter in feature set 8. To evaluate the impact of user speed on each of the aforementioned feature sets, we devise sets 3, 5, 7 and 9.

A comparison of the feature sets intelligently derived using Sobol analysis is given in Table 5.3. The analysis of feature sets 1, 2, 4, 6 in terms of area under receiver operating characteristic (AUROC) curve and area under precision recall curve (AUPRC) highlights that the input feature set 4 that includes four most important features

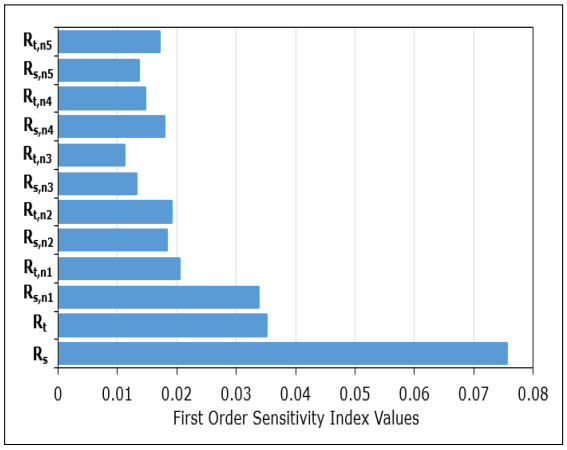


Fig. 5.5: Sobol sensitivity analysis of the input features used for HOF prediction.

Table 5.3: Different Set of Input Features and Their Effect in HOF Prediction

Set #	Feature Vector	AUROC	AUPRC
1	$\{R_s\}$	0.877	0.635
2 [135]	$\{R_s,R_t\}$	0.875	0.628
3	$\{S,R_s,R_t\}$	0.884	0.638
4	$\{R_s,R_t,R_{s,n1},R_{t,n1}\}$	0.919	0.643
5	$\{S,R_s,R_t,R_{s,n1},R_{t,n1}\}$	0.920	0.637
6	$\{R_s,R_{s,n1},R_{s,n2},R_{s,n3},R_{s,n4}, R_{s,n5}, R_t,R_{t,n1},R_{t,n2},R_{t,n3}, R_{t,n4},R_{t,n5}\}$	0.916	0.635
7	$\{S,R_s,R_{s,n1},R_{s,n2},R_{s,n3},R_{s,n4}, R_{s,n5}, R_t,R_{t,n1},R_{t,n2},R_{t,n3}, R_{t,n4},R_{t,n5}\}$	0.928	0.643
8	$\{R_s,R_t,G_s,G_t\}$	0.947	0.698
9	$\mathbf{\{}S,R_s,R_t,G_s,G_t\mathbf{\}}$	**0.951**	**0.713**

performs better than the feature sets 1, 2 and 6. The better performance of feature set 4 compared to feature set 6 shows that the HOF prediction does not improve by just adding more interferers as input features. Meanwhile, adding user speed as input features (i.e., sets 3, 5, 7) showed improvement in the AUROC and AUPRC performance compared to their counterpart sets 2, 4, and 6, respectively.

To this end, we devise feature set 8, where G-Factor is used as an input feature instead of raw interference. G-factor combines the impact of all the interferers in one input feature as described in eq. (5.3). Results in Table 5.3 show that this domain knowledge aware smart feature improves the AUPRC of feature set

8 compared to other aforementioned feature sets. Finally, the speed of the users is also included as a new input feature to generate feature set 9. Results show that the input feature set 9 with user speed and G-Factor yields better performance compared to all other feature vectors with AUROC and AUPRC of 0.951 and 0.713, respectively.

Results from the feature engineering analysis led to the formulation of the optimal input feature vector $\mathbf{V}_o$ expressed as:

$$\mathbf{V}_o = \{S, R_s, R_t, G_s, G_t\}, \tag{5.8}$$

where S corresponds to user speed, R and G correspond to RSRP and G-Factor, respectively with subscripts s and t correspond to the serving and target inter-frequency BSs, respectively.

5.5 Handover Failure Mitigation Algorithm and Performance Evaluation

This section focuses on the presentation of the HOSR improvement algorithm to complete the second sub-solution of the TORIS framework. Moreover, we also discuss the method of evaluation to measure the effectiveness of the proposed algorithm.

5.5.1 Tx Power Tuning based HOSR Improvement Algorithm

The insights gained from the HOF prediction model discussed in Section 5.4, we conclude that HOFs occur due to certain combinations of RSRP of the source and target BSs. Modification of soft parameters such as CIO is one way to improve HO performance. Handover can happen earlier or later with CIO tuning and hence changing the RSRP conditions of source and target BS during handover. However, CIO cannot be tuned for every user and can result in instances of HOFs for different

set of users performing HO between the same source and target BS. To solve this problem, we propose to mitigate HOF for each type of HO instance by changing the Tx power of the serving and target BS instead of tuning soft parameters such as CIO [1].

Algorithm 1 presents the proposed method to mitigate HOF by varying the Tx power of the source and target BSs. In each handover request made by the UE, the reported measurements are fed into the trained AI classifier. For each input combinations, the AI model detects if the requested HO will fail or not. In case of a handover success prediction, the source BS commence with the handover to the target BS. However, if there is a predicted HOF due to the received measurement, the source BS incrementally increases its Tx power. Increasing the Tx power is the method of choice since decreasing the Tx power might lead to further degradation of signal condition and quality, which will lead to more chances of HO failures.

The increment in source Tx power is chosen from source BS increment vector $\vec{I_s}$ = [0, δ_s, $2\delta_s$,..., I_s^{max}], where δ_s and I_s^{max} are the increment value and maximum power increase for source BS, respectively. In each iteration, the new source Tx power T_s^{new} is computed by adding a value of increment from $\vec{I_s}$ in existing source Tx power T_s. Increasing the Tx power of source BS is given preference over the target BS because it has the highest impact on HOF as shown in Fig. 5.5. For each increment made, the algorithm feeds back the new measurement combination to the HOF prediction model. The model then checks if the new combination results in HOF or not. Once the maximum power increment in the source BS reaches the limit and the HOF prediction model still detects unsuccessful HO, the algorithm increments the Tx power of the target BS. The increment in target Tx power is chosen from target BS increment vector $\vec{I_t}$ = [0, δ_t, $2\delta_t$,..., I_t^{max}], where δ_t and I_t^{max} are the increment value and maximum power increase for target BS. In each iteration, the new target Tx power T_t^{new} is computed by adding a value of increment from $\vec{I_t}$ in existing target

Tx power T_t. The Tx power of source and target BS are increased to T_s^{new} and T_t^{new}, respectively only when the AI model indicates a successful HO. Similar to CIO values, the increment and limit for maximum change in the Tx power can be controlled by the operator. In this chapter, we use 1dB value for δ_s and δ_t and 3dB value for I_s^{max} and I_t^{max}. HO process commences only after a HOS prediction by the AI model or the maximum limit for both source and target Tx power variation is reached.

Since the increment in Tx power only occurs momentarily for a specific signaling message, the probability of two or more neighboring base stations to transmit handover signaling messages at the exact time instant is very low. However, for a very rare case wherein two or more base stations transmit signaling message and increase the Tx power at the same time, the performance of the proposed algorithm might degrade due to potential momentary increase in interference and decrease in the SINR. In addition, the increase in Tx power for one TTI will impact other UEs in the network only during this TTI. The signal strength of the users connected to the base station increasing the Tx power will improve but the SINR of the users connected to neighboring base stations (operating at the same frequency) will decrease. Similarly, the momentary increase in the Tx power is not enough to trigger unnecessary handover due to its much shorter duration (1ms) compared to handover TTT (mostly more than 40ms). Therefore, the impact on other UEs in the system will be for one TTI and very sparse in time.

5.5.2 Handover Failure Mitigation Scheme Evaluation

Inaccuracies in HOF detection model creates two scenarios. First, handover failure mitigation may not activate due to incorrect prediction of HOF as HOS. This shortcoming is captured by evaluating the number of uncompensated HOF. These instances refer to the cases when HOF are not predicted or resolved correctly. Second

Algorithm 1 Tx Power Tuning-Based Handover Failure Mitigation

```
for each handover request do
    (a) HOF prediction using trained AI models;
    if HOF is true then
        for i in I_t = [0, δ_t, 2δ_t,..., I_t^max] do
            (b) T_t^new = T_t + I_t[i];
            for j in I_s = [0, δ_s, 2δ_s,..., I_s^max] do
                (c) T_s^new = T_s + I_s[j];
                (d) HOF prediction using trained AI model with T_t^new and T_s^new;
                if HOF is false then
                    break both loops;
                end
            end
        end
    end
end
```

scenario happens when HOF mitigation is falsely activated due to wrong prediction of potential HOS as potential HOF. To capture this, we have measured the number of false compensations, which are the recorded number of instances wherein the algorithm kicks in to compensate a handover request unnecessarily. False compensations in addition to delay in HO can lead to waste in power resources due to the unwanted increase in Tx power. A slight instantaneous increase in interference on the particular signaling messages, to nearby users if they are going through HO at the same time with neighboring BSs, is another cost of false compensation. Since increase in Tx power is done for only certain TTIs, the impact of false compensation rate can be determined by the number of false compensations affected TTIs over the total simulation TTI (i.e., 15000ms).

We measure the performance of proposed HOF minimization scheme by defining a metric which captures both uncompensated HOF and false compensations. Firstly, the number of uncompensated HOFs (Δ) due to errors in HOF prediction can be written as:

$$\Delta = \Theta_t - \Theta_c, \tag{5.9}$$

where Θ_t represents the total number of HOFs and Θ_c shows the total number

of compensated HOFs. Since ideally, all HOF should be resolved, the value of Δ should be as small as possible. Secondly, false compensation (Γ) also causes negative impact on the network performance and thus should be as minimal as possible. Therefore, the combined failed compensation and false mitigation score Υ is calculated as:

$$\Upsilon = (\alpha \times \Delta_{norm}) + [(1 - \alpha) \times \Gamma_{norm}], \tag{5.10}$$

where Δ_{norm} and Γ_{norm} are the normalized values of uncompensated HOFs Δ and false compensations Γ, respectively, α corresponds to the operator-defined weight with values from 0 to 1. Operator-defined means that the network operator can choose the value of α depending on their preference. α can be used to set the priority between minimizing uncompensated HOFs and false compensations. For instance, a high value of α will give more priority to uncompensated HOF minimization compared to potential false compensation. Similarly, it can be set to 0.5 to give equal priority to uncompensated HOFs and false compensation. For different values of α, best augmentation technique can be evaluated as the one having the least value of false mitigation score Υ.

5.6 Results and Discussion

5.6.1 Synthetic Data Evaluation Results

Statistical Evaluation Results

Tabular synthetic data is generated by SMOTE, CLBN, GAN, BL-SMOTE, BL-CLBN and BL-GAN. Each technique produces five column vectors, and each column corresponds to an input feature similar to the real data. To overcome the complexity of comparing the real and synthetic data using each column individually, we compare them using a single parameter that captures the likelihood of the real and synthetic data as a whole. We achieved this goal through a dimensionality reduc-

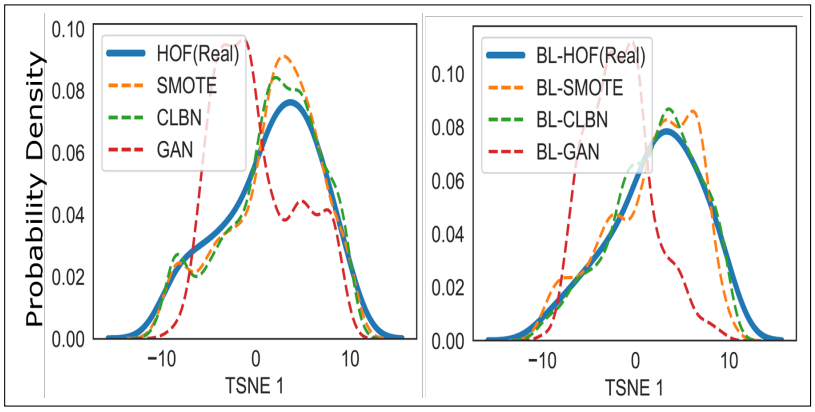


Fig. 5.6: PDF comparison between real and synthetic data using different data augmentation techniques.

tion technique called t-Distributed Stochastic Neighbor Embedding (t-SNE). This method transforms the input vector with 5 dimensions into a single t-SNE component. Fig. 5.6 shows the probability distribution function (PDF) of the single t-SNE component of real and synthetic data. From the results in Fig. 5.6, we see SMOTE and CLBN synthetic data follow the distribution of the real data very closely. However, distribution of data produced using GAN varies from the distribution of the real data. When using the borderline technique, results show that BL-SMOTE and BL-CLBN maintain the close proximity to the real data PDF. On the other hand, the distance of BL-GAN from the real distribution becomes even larger.

Pairwise Correlation Comparison Results

To avoid cluttering, Fig. 5.7 shows only the best and worst correlation which are from SMOTE and BL-GAN, respectively. Results from Fig. 5.7 show that the correlation between variables in real and synthetic data set is maintained using SMOTE. This is indicated by smaller values of pairwise correlation for SMOTE. However, pairwise correlation difference is high for BL-GAN, i.e, the relationship between source RSRP and source G-Factor is 0.67 as well as the correlation between target RSRP and target G-Factor is 0.36. These results show that BL-GAN does not yield synthetic data in which the relationship between the input features mimics

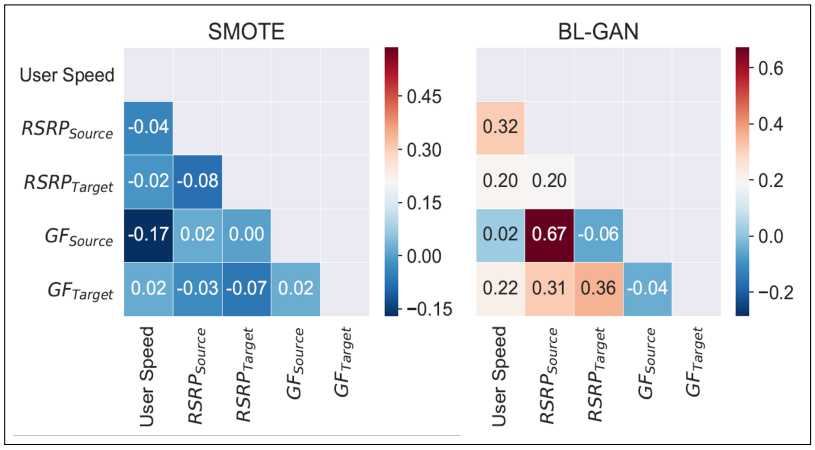


Fig. 5.7: Best and worst pairwise correlation difference comparison.

Table 5.4: Summary of Evaluation Comparison Between Different Data Augmentation Techniques

	EMD (ζ)	**Mean PCD (μ)**
SMOTE	0.325	0.047
BL-SMOTE	0.365	0.101
CLBN	0.312	0.054
BL-CLBN	0.230	0.106
GAN	2.040	0.177
BL-GAN	3.953	0.238

that in real data.

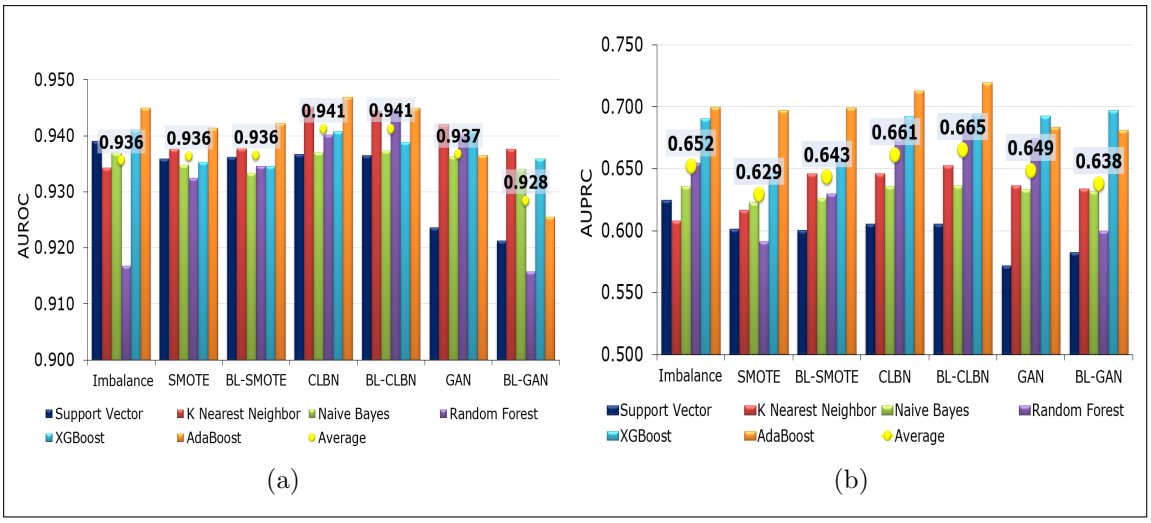


Fig. 5.8: AUROC (a) and AUPRC (b) comparison between different augmentation techniques using different AI algorithms.

Synthetic Data Evaluation Summary

Summary of results evaluation in Table 5.4 show that SMOTE and CLBN, together with their BL counterpart, generate data with good quality in terms of approximating the distribution and feature correlation with respect to the real data. The corresponding small values of ζ for these techniques signify that the marginal distributions of real and synthetic data sets are very close to each other. Similarly, the small values of μ indicate the unchanged relationship between the input features. On the other hand, the quality of data generated by GAN and BL-GAN is poorer compared to other techniques. The poor performance of GAN can be attributed to the fact that it is based on deep learning, which requires a lot of training data unlike SMOTE and CLBN. This evaluation shows that GAN is not suitable for addressing imbalance data problem with few training samples.

Ideally, data with poor quality should be discarded and should not be utilized for AI model training. Such kind of data might mess up with model training instead of improving them. However, to show the impact on model training, we keep the data generated by GAN to observe how it affects the model performance.

5.6.2 Handover Failure Prediction Results

To evaluate the effectiveness of the synthetic data generation in improving the performance of our HOF prediction model, we analyze the AUROC and AUPRC of different AI algorithms with and without data augmentation. Fig. 5.8(a) shows the AUROC comparison of different AI models. We observe that data augmentation techniques such as SMOTE and BL-SMOTE do not provide a significant improvement in terms of AUROC. AUROC of the model trained on the data without augmentation has an average of 0.936, which is similar to the average AUROC of models trained with augmented data from SMOTE and BL-SMOTE. Meanwhile, training the models using data generated from CLBN and BL-CLBN show improve-

ment with an average AUROC of around 0.941 for both. When models are trained using data from GAN, average AUROC is again comparable when using real data only. However, using BL-GAN, AUROC drops to 0.928. Results also show that augmentation technique is particularly beneficial for Random Forest and KNN.

Results in Fig. 5.8(b) shows the variations in the AUPRC for different models trained with imbalance data set and balance data set using different augmentation techniques. Evaluation of the models which are trained using SMOTE and BL-SMOTE indicates that these approaches cause degradation in the performance of the classifiers. From AUPRC of 0.652 without data augmentation, AUPRC drops to 0.629 and 0.643 with application of SMOTE and BL-SMOTE respectively. Particularly, SMOTE and BL-SMOTE cause a significant decline in the performance of Random Forest and XGBoost, which drags down the total average AUPRC. Meanwhile, models which utilize data generated from CLBN and BL-CLBN for training show AUPRC improvement. Both methods seem to work well in improving AUPRC of almost all the AI classifiers especially KNN and Random Forest. Improvement in the performance of AdaBoost is also notable using CLBN-based augmentation techniques. However, once again, GAN and BL-GAN fail to enhance the overall AUPRC performance of the models. These techniques only improve the performance of KNN while the rest of the classifiers degrade. These results mirror the poor quality of data generated using GAN.

These results highlight the importance of thorough synthetic data evaluation. Without data evaluation, models train with GAN-based data augmentation seem to perform fine. In fact, utilizing GAN even improved the overall AUROC of the models. These results reveal that looking solely on the effect of data augmentation techniques on prediction performance of the model is tricky and extra precaution is needed when using data augmentation techniques such as GAN. Results also show that different AI models have different sensitivities to the quality of data fed into

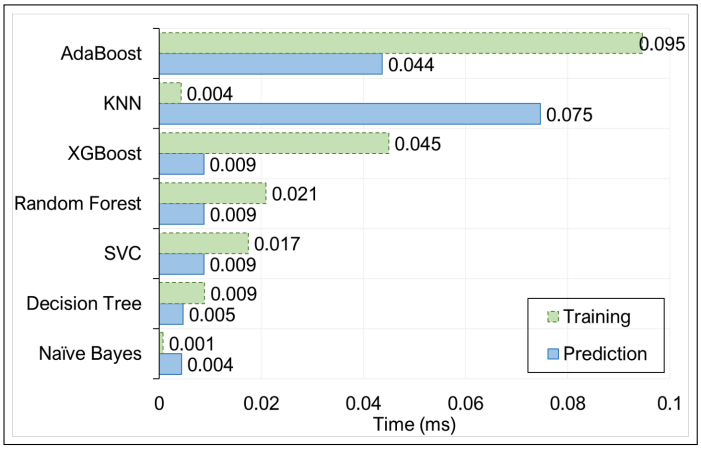


Fig. 5.9: Time complexity comparison of different AI models.

them. For instance, the performances of Support Vector Classifier and AdaBoost are severely affected by the poor quality of data while others are not.

We also evaluate the time complexity of the AI models we used for HOF prediction. Time complexity is an important performance metric especially for time sensitive problems such as HOF prediction. Fig. 5.9 shows the training and prediction time of the AI models normalized against the number of data points in training and test set. In terms of training time, results show that Naïve Bayes and KNN are the fastest while AdaBoost is the slowest. Meanwhile, with regards to the prediction time, again, Naïve Bayes has the shortest time while KNN has the longest followed by AdaBoost. However, although faster, the prediction accuracy of algorithms such as Naïve Bayes is significantly worse than AdaBoost. Moreover, the prediction time of AdaBoost, although higher than the others, is significantly less than the least possible TTI of 5G communication. 5G design includes scalable TTI, corresponding to slot duration between 62.5µs and 1ms. In 3GPP Release 16, 120kHz is the maximum sub-carrier spacing allowed for data communication which makes the lowest achievable slot duration equal to 125µs, far longer than the 44µs prediction time of AdaBoost. Thus, even with high complexity, we use AdaBoost to leverage the superior performance in detecting HOFs.

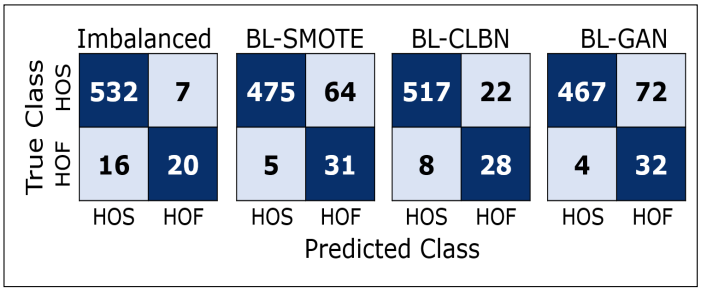


Fig. 5.10: Confusion matrix comparison between different data augmentation techniques using Adaboost

5.6.3 Shift in the Decision Boundary after Class Balancing

The shift in the decision boundary towards the minority class after data augmentation poses a trade-off between detection of more HOFs and the overall accuracy of the model. This trade-off reflects with the variation in the confusion matrix using different data augmentation techniques as shown in Fig. 5.10. The confusion matrix results are shown only for the best performing classifier, AdaBoost, using the validation data set. Results reveal that Adaboost model correctly classify 20 out of 36 HOFs when it is trained on the imbalance data set with only 7 HOS misclassified as HOF. Meanwhile, the number of correctly detected HOFs increase to 31 with the utilization of BL-SMOTE. However, the number of misclassified HOS as HOF increase from 7 to 64 alongside. Similarly, the HOF detection performance of the model trained with synthetic data from BL-GAN increases in expense of more HOS misclassifications. On the contrary, BL-CLBN displays a fair detection rate of HOF without causing too many misclassifications of HOS as HOF. BL-CLBN correctly predict 28 out of 36 HOFs with 22 HOS misclassified as HOF.

5.6.4 Handover Failure Mitigation Results

The effectiveness of the proposed HOF mitigating scheme hinges on the performance of the HOF prediction model. Having analyzed the performances of various HOF prediction models using different class balancing techniques, in this sub-section we

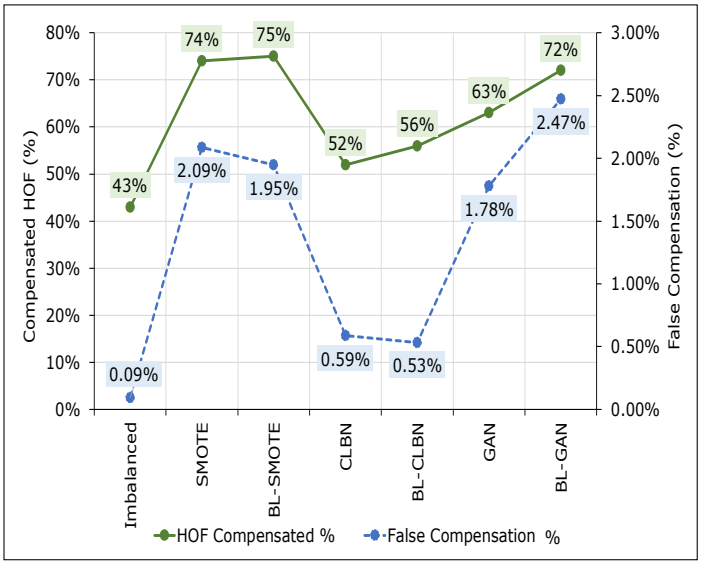


Fig. 5.11: Handover failure mitigation performance using different data augmentation techniques.

discuss the results of handover failure mitigation algorithm.

Fig. 5.11 shows the performance of the HOF mitigation algorithm using the percentage of HOF resolved with corresponding false compensation rate. Results show that without any data augmentation, around 43% of the HOFs are mitigated. However, this number increases when data augmentation is incorporated. For instance, the highest number of HOFs are detected using SMOTE and BL-SMOTE. Thus, more HOFs are compensated by the HOF mitigation algorithm which is around 30% more compared to imbalanced data set. However, it is notable that using SMOTE and BL-SMOTE will also lead to a higher false compensation rate of around 2.09% and 1.95%, respectively compared to 0.09% for imbalanced data set. A more conservative choice is either CLBN or BL-CLBN with only 0.59% and 0.53% false compensation rate, respectively. At around 56%, the performance of BL-CLBN in terms of resolving HOFs is better compared to CLBN at 52%.

Table 5.5 shows the failed compensation and false mitigation score (Υ) given in eq. (5.10) with different values of α. We observe that a model trained on BL-CLBN generated synthetic data set performs the best when equal priority ($\alpha = 0.5$) is given for uncompensated HOF and false compensation. Meanwhile, model utilizing

Table 5.5: Failed Compensation and False Mitigation Score Υ for Different values of α

	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.9$
Imbalanced	0.500	0.900	**0.100**
SMOTE	0.435	0.113	0.757
BL-SMOTE	0.389	**0.078**	0.701
CLBN	0.464	0.670	0.259
BL-CLBN	**0.388**	0.550	0.225
GAN	0.535	0.395	0.674
BL-GAN	0.541	0.174	0.908

data using BL-SMOTE has the best performance when more weight is given to minimize false compensation rather than failed compensation ($\alpha = 0.1$). Model trained without data augmentation performs the best when more weight is given to minimize failed HOF compensation ($\alpha = 0.9$). These results show the capability of the proposed HOF mitigation approach to provide leeway to the network operators to set priorities as well as select the appropriate data augmentation technique.

Thus far, it is evident that the current solution offers two mechanisms to benefit the insights gained from the false compensation analysis: (1) A tunable weight α is incorporated wherein a vendor or operator can choose to minimize the false compensation at the expense of missing more handover failure detection as discussed in Section 5.5, and (2) The results in Fig. 5.8, Fig. 5.11, and Table 5.5 provide insights for operators to choose an AI and augmentation method combo that offers that aligns with operators' priority for minimizing HOF failures or minimizing unwanted momentary increase transmission power. These methods offer an offline strategy to minimize the false compensation at the expense of less HOF mitigation.

Finally, to verify the effectiveness of the proposed TORIS framework, we compare its performance with the state-of-the-art alternative, i.e., CIO tuning algorithm [1], for improving HOSR. The algorithm used by [1] is implemented in SyntheticNET [27]. For a fair comparison, the same network deployment, number of users, user speed, user mobility as well as the same handover parameters are used. Fig. 5.12 presents the findings of the comparison study. The results show that CIO-based

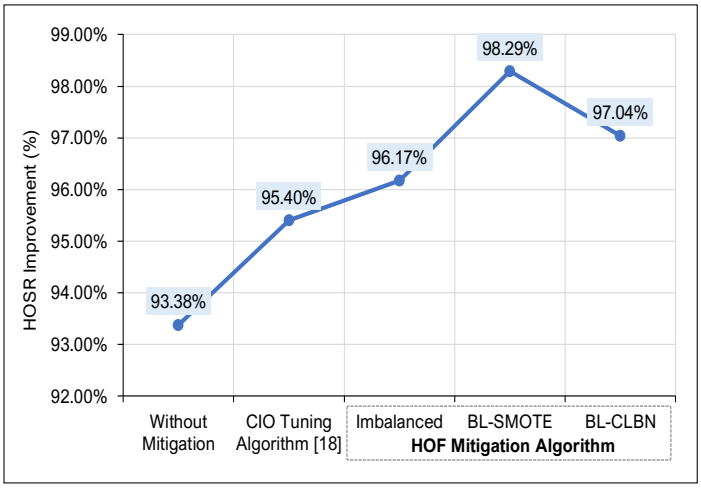


Fig. 5.12: HOSR ξ improvement comparison between algorithm used in [1] and proposed HOF mitigation algorithm.

tuning to mitigate HOF only improves the HOSR by around 2%. In contrast, the proposed TORIS improves the HOSR from 93.38% to around 96%-98%. It is worth noting that comparison in terms of false mitigation could not be done as proposed solution in [1] does not involve variation in Tx power.

5.7 Conclusion

This chapter proposes an AI powered Tx power tuning-based solution to improve inter-frequency handover success rate, called TORIS. TORIS consists of two distinct components. 1) An AI-based model to accurately predict HOF despite of imbalanced training data. 2) A serving and target cell Tx power tuning algorithm that feeds on the prediction of the HOF model to minimize the HOF and thus improve HOSR. First challenge in creating an HOF model is selection of training feature. We leverage domain knowledge to devise novel set of training features that outperform the feature combinations used in literature. The second key challenges in creating an AI-based HOF prediction model comes from the imbalanced training data where the number of HOF instances in real network are far less than the HOS. We show that this data imbalance can lead to poorly performing model even with the state-of-the-art classifiers. We solve this problem by thoroughly investigat-

ing a range of data augmentation techniques and their novel combinations such as SMOTE, CLBN and GAN. Leveraging the idea of using the borderline, we devise a hybrid CLBN and GAN called BL-CLBN and BL-GAN. Results show that the former performs best in terms of AUROC and AUPRC of the HOFs model trained on the augmented data. Extensive synthetic data evaluation shows that the quality and validity of data generated by GAN is poorer in comparison with data generated using other techniques due to its inherent dependency on large training data. Both SMOTE based and GAN based approaches tend to shift the models' decision boundaries toward detection of more HOFs at expense of misclassification of HOSs as HOFs. On the other hand, CLBN based HOF prediction models predict more HOFs compared to imbalanced data trained HOF models while keeping in check the amount of misclassified HOSs. Upon quantifying the effect of misclassifications, BL-CLBN shows superior performance in mitigating HOFs when equal priority is given to HOF prediction and HOS misclassifications. Finally, we compare the performance of TORIS with state-of-the-art HOSR enhancement scheme that uses CIO tuning instead of Tx power tuning. Results show that proposed TORIS scheme outperforms the state-of-the-art by significant margin in improving the HOSR i.e., CIO based scheme raises HOSR from 93.38% to only 95.40%. In contrast TORIS can raise HOSR all the way to 98.29%. The gain of the proposed TORIS can be attributed to novel feature selection, data balancing and parameter turning algorithm that addresses the root cause of the HOF i.e., Tx power misconfiguration instead of tuning CIO.

CHAPTER 6

Introduction of a Novel Handover Parameter for Emerging AI-enabled Cellular Networks

6.1 Introduction

While the proposed solution presented in Chapter 5 can help improve the mobility robustness in emerging cellular networks, it still relies on the current standardized parameters such as event A3 offset, TTT, and CIO. However, despite its usefulness, the inflexibility of these parameters including CIO still poses several limitations. As CIO is configured per neighbor relation (i.e., from cell 1 to cell 2), it has limited influence over the specific user equipment (UE) that will be impacted by CIO adjustment. Therefore, modifying the CIO relationship between neighboring cells may result in favorable outcomes for some users, while adversely affecting others. For example, changing the CIO relationship between cells may result in unintended handovers to even stationary users. Furthermore, adjusting CIO to manage load balancing could potentially lead to congestion in the target cell, due to the limited control over the selection of UEs for handover. These limitations of CIO suggests that it may not be sufficient to meet the ever-growing demand for mobility in future networks.

To overcome these limitations, we introduce a novel handover parameter, referred to as User Individual Offset (UIO), to complement the established and standardized CIO parameter. Although both CIO and UIO can influence handover decisions, CIO is configured per cell relation, while UIO is tailored to each user. This approach offers greater flexibility in assigning handover parameter values, providing more precise control over handover operations. The UIO value for each user is unique

and determined by factors such as user velocity, trajectory, mobility pattern, and service requirements. By incorporating these factors, this method can optimize handovers for each user, thereby improving the overall network performance.

6.1.1 Related Work

Several studies regarding more tailored and customized parameter settings do exist. The related studies can be divided into two major categories. The first category are the studies proposing to set the handover parameters such as HOM and TTT based on grouping users with similar behaviors. The second consists of technologies where individual users are assigned different handover parameters.

To improve the handover performance of users, one strategy entails user classification (i.e., using various machine learning and data mining techniques) followed by the customization of handover parameter settings to meet the specific requirements of each user class. Researchers in [156] utilized t-distributed stochastic neighbor embedding (t-SNE) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to group users based on their reported RSRP from the source and neighboring base stations. The authors in [156] then adjusted the handover parameters based on the user groups, resulting in reduced call drop rates, decreased ping-pong handover rates, and an improved user experience. In another study [157], the authors used the Analytic Hierarchy Process and Technique for Order Preference by Similarity to Ideal Solution (AHP-TOPSIS) approach to determine the optimal base station for serving users. They also grouped the users based on their application needs, such as delay sensitivity and speed sensitivity, and used Q-learning to determine the ideal TTT and hysteresis settings to minimize handover issues like ping-pong, late handover, and early handover. In [158], the authors employed machine learning techniques to identify unique patterns in the RSRP values reported by users during the handover process. Customized handover parameters

were then applied to in-building base stations to improve the user experience. Similarly, in [159], users experiencing frequent handovers were classified as fast-moving or ping-pong users based on their mobility behavior, and tailored handover parameters were applied accordingly. In [160], a fuzzy logic-based approach was used to tune the handover hysteresis based on the user's velocity and radio channel quality. In [161], an algorithm was proposed to dynamically optimize handover parameters such as hysteresis and TTT based on user location, RSRP, signal to interference and noise ratio (SINR), and speed. The proposed algorithm aims to minimize handover failures and improve the success rate. Another proposed method in [162] used an auto-tuning optimization (ATO) algorithm to modify handover parameters based on UE RSRP and speed, with the goal of reducing handover failures. Finally, in [163], the authors presented an adaptive handover mechanism that used a Kalman filter to estimate the future signal quality, state–action–reward–state–action (SARSA) based reinforcement learning to choose the target cell for the handover, and ϵ-greedy policy to adjust the TTT and hysteresis.

Although most of the existing literature on handover management focuses on the first category, there are also studies that investigate individualized handover optimization. For instance, in [164], the authors proposed an algorithm that dynamically adjusts handover settings for each user based on a weight function that considers factors such as SINR, cell load, and UE speed. By doing so, the proposed algorithm improved the RSRP, reduce ping pong handovers, and minimize radio link failures (RLFs).

Although the studies discussed earlier have proven effective in enhancing user mobility key performance indicators (KPIs), they have limitations in terms of customization. Specifically, methods under the first category only categorize users into clusters before applying handover parameter values, which restricts their flexibility. Additionally, updating parameters such as TTT and HOM whenever user behavior

changes, as proposed in [161–163], would significantly increase signaling overhead. On the contrary, our proposed approach using UIO only needs to be transmitted once or updated at each handover occurrence, making it a more efficient solution. It is also worth noting that none of the relevant publications proposed new parameters in either category, relying instead on existing ones such as HOM and TTT. As a result, these methods do not provide further optimization flexibility for cellular networks, unlike UIO.

6.1.2 Contributions

To address the limitations of the current solutions in literature, we present UIO, a new handover parameter that can be customized on a per user basis. The main contributions of this work can be summarized as follows:

1. This chapter introduces a new parameter called UIO, which can be customized according to various UE attributes, such as speed, trajectory, and service requirements. By offering this level of flexibility, UIO provides a new dimension for optimizing network performance beyond the constraints of traditional handover parameters like CIO.

2. The inclusion of this extra dimension adds a layer of complexity to the task of optimizing the network. If determining the optimal CIO is already challenging, the task becomes even more difficult when considering that individual users require personalized sets of UIO. To tackle this issue, we have put forward several approaches for calculating UIOP. These methods encompass a range of techniques, such as leveraging machine learning, reinforcement learning, graph neural networks, and digital twin technology, among others.

3. This chapter explores the effectiveness of various AI techniques for estimating KPIs from different UIO values, user speed and direction. Combining these

techniques with heuristics, we were able to develop a solution that efficiently calculates UIO. This solution has been integrated into a framework that assigns specific UIO values to each user. By utilizing our domain knowledge, we were able to narrow down the search space and only calculate the necessary UIO values.

The remainder of this chapter is structured as follows. Section 6.2 provides a more detailed explanation of the underlying principles and mechanisms of the UIO parameter. In Section 6.3, we present the proposed framework for calculating the UIO values, including the problem formulation, simulation setup, and data generation. In Section 6.4, we analyze the impact of UIO on various handover-related KPIs. Section 6.5 presents the results of both machine learning and heuristic search-based approaches for calculating the UIO values. Finally, in Section 6.6, we conclude the chapter and discuss future work.

6.2 UIO Introduction

UIO is an innovative HO parameter that is closely related to the current 3GPP standardized parameter known as CIO. While CIO lacks flexibility at the user level, UIO is a user-specific parameter that can control the handover behavior of each user individually. This enables the selection of UIO values that account for diverse user behavior such as speed, direction, and service requirement. The value of UIO, similar to CIO, is represented in dB and can be positive or negative. Prior to initiating a handover event, the measured signal (e.g., RSRP or RSRQ) is modified by the UIO value. If a positive UIO value is used, the UE reports to the base station that the measured signal value is higher than the actual value (i.e., actual value plus UIO). Conversely, a negative UIO value is used to indicate that the measured value is lower than the actual value (i.e., actual value minus UIO).

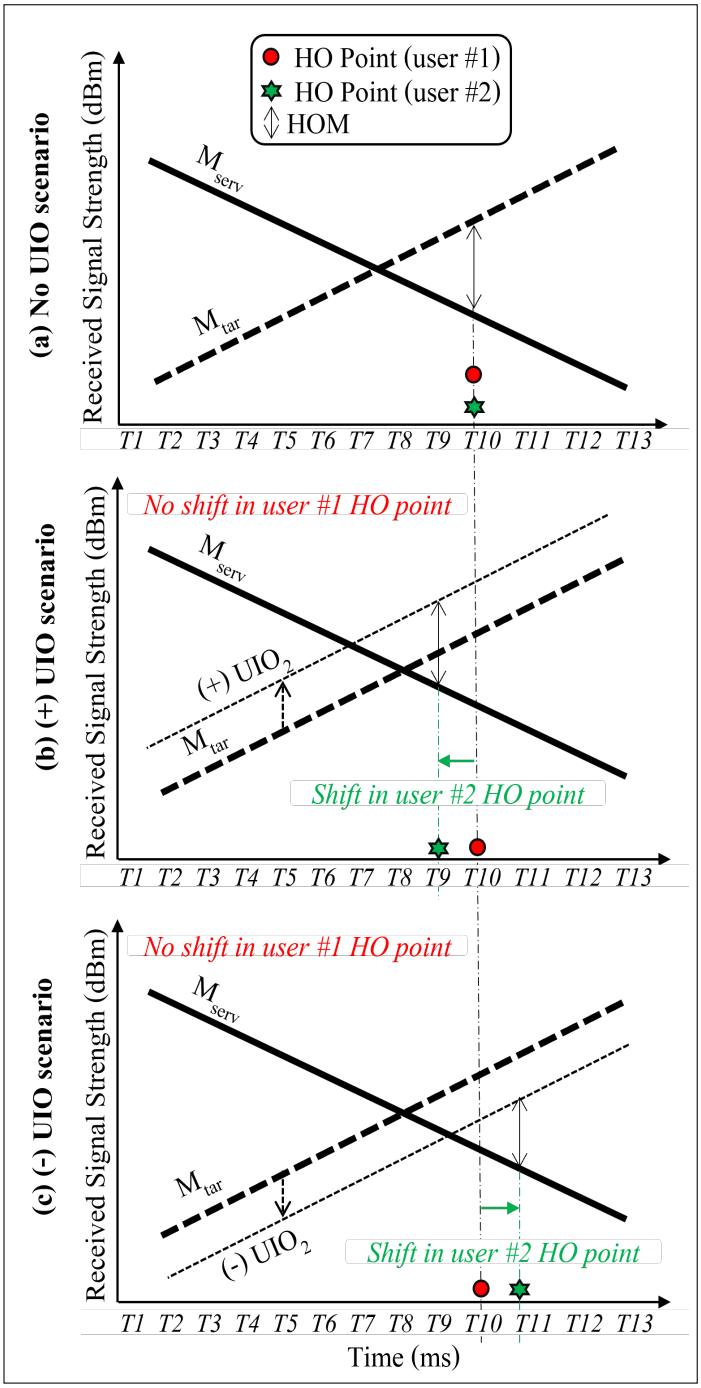


Fig. 6.1: Illustration of how UIO controls the handover behavior of the users.

To illustrate the impact of UIO on handover, we provide an example that uses event A3 as a trigger for handover. Event A3 is a mechanism that ensures the UE will perform a handover to a base station (BS) with a better signal condition by controlling the mobility decision using an offset parameter. This event is triggered when the neighboring BS signal becomes better than the serving BS signal by an offset value, considering the effect of hysteresis, cell individual offset, and frequency offset. The event report is sent when the following condition remains true for a duration set by the time-to-trigger (TTT) parameter:

$$M_{tar} + O_{tar,freq} + O_{tar,cell} - hyst > M_{serv} + O_{serv,freq} + O_{serv,cell} + off \quad (6.1)$$

where the left side of the inequality represents the parameters related to neighboring cells, including M_{tar}, which represents the measured value of RSRP or RSRQ, $O_{tar,freq}$, which denotes the frequency-specific offset, $O_{tar,cell}$, which indicates the cell-specific offset, and hysteresis, denoted by $hyst$. On the right-hand side, we have the parameters related to the serving base station, with the addition of the offset (off) parameter. It is worth noting that $O_{tar,freq}$ and $O_{serv,freq}$ are used only when event A3 is configured for inter-frequency handover, whereas they are not used for intra-frequency handover.

Considering intra-frequency handover, eq. (1) becomes:

$$M_{tar} + O_{tar,cell} - hyst > M_{serv} + O_{serv,cell} + off \quad (6.2)$$

By combining the $hyst$ and off parameters into a single parameter called handover margin (HOM), and the $O_{tar,cell}$ and $O_{serv,cell}$ parameters into a single CIO parameter, eq. (2) can be simplified as follows:

$$M_{tar} + CIO > M_{serv} + HOM \quad (6.3)$$

When the HOM is fixed to a constant value, it becomes evident that a positive

CIO applied from the serving to the target cell facilitates the fulfillment of the event A3 condition, leading to a faster handover trigger from the source to the target BS. Conversely, a negative CIO makes it more challenging to meet the event A3 condition, thereby delaying the triggering of the event and, consequently, the handover.

Eq. (3) represents the current 3GPP standardized event A3 evaluation for handover. All UEs connected to the source BS utilize this evaluation criteria. This nuance of the current handover standards makes user-level control of handover behavior challenging. In this regard, UIO can provide a new level of flexibility to meet this challenge. The integration of UIO with the event A3 condition is expressed as follows:

$$M_{tar} + CIO + UIO_x > M_{serv} + HOM \tag{6.4}$$

where the subscript x denotes that this new attribute can be customized separately for each user. By incorporating this additional parameter, each user can have a distinct handover evaluation condition, providing the network with greater control over the handover behavior of individual UEs.

The ability of UIO to control the handover (HO) behavior of individual users is illustrated in Figure 6.1. Consider two users, user #1 and user #2, both camped on a similar BS and traveling at the same speed and direction at the same time. As these users move away from one BS and closer to another, the receive signal level (M_{serv}) of the serving BS decreases while the signal level for the target BS (M_{tar}) increases. To ensure continuous service for UEs, the source and target cells must coordinate to aid in handover. When the received signal strength from the target BS exceeds that of the source BS, considering HOM, the serving BS transfers the UEs to the target BS. Figure 6.1(a) depicts the case where no UIO is implemented. Without UIO, the handover points for both UEs occur at time (i.e., $T10$). However, UIO

can be selectively applied to individual users to modify their handover behavior. In this example, UIO is applied only to user #2, adjusting its handover point without influencing the handover behavior of other users, such as user #1. With a positive UIO towards the target cell, as depicted in Figure 6.1(b), the transition point for UE #2 changes to $T9$, resulting in an earlier handover. Conversely, when a negative UIO is imposed on the target cell, as shown in Figure 6.1(c), the handover for both UE #2 shifts from $T10$ to $T11$, resulting in a later handover.

6.2.1 Limitations of CIO

The closest technology to UIO is the CIO, however, UIO has multiple advantages over this current state-of-the-art. Firstly, UIO allows the parameter adjustment on individual user behavior, such as user speed, trajectory, serving base station, target base station, and service requirements etc. This individualized nature of the UIO permits the improvement of key performance indicators (KPIs) for all users, whereas CIO tends to degrade KPIs for some users. It is challenging to guarantee the overall network and service performance when using or adjusting the CIO. For instance, changing the CIO relationship between neighboring cells could result in heavy traffic on the target cell because we cannot choose which UEs will execute handover. With UIO, we can specify which UEs can perform handovers during load balancing and have a greater degree of control over the number of users that can be transferred from one cell to another. This tendency is further explained and verified based on the preliminary results we presented in the preceding section. Secondly, UIO has the potential to eliminate the conflict between self-organizing network (SON) solutions such as mobility robustness optimization (MRO) and mobility load balancing (MLB) as these SON functions aim at adjusting CIO with opposite values.

To illustrate the potential detrimental impact of tuning CIO values, we are running an experiment to test its impact on static users. Using our domain expertise, we

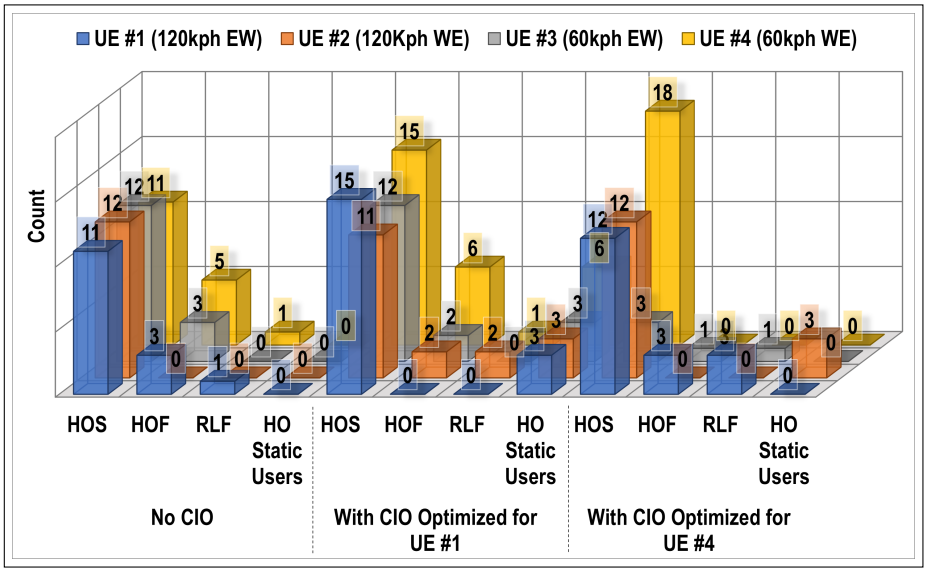


Fig. 6.2: Impact of CIO on static users.

assign CIO to generate four network scenarios: a network with default CIO (0dB is used in all neighbor relations), a network with CIO optimized for user #1, and a network with CIO optimized for user #4. Optimizing CIO has an effect on the handover performance of mobile and static users, as shown in Figure 6.2. Using the default configuration (i.e., no CIO), we observe that users #1, #3, and #4 suffer handover issues, including handover failures and RLFs. Since there is no CIO, it makes sense that there is no handover for the static users. Then, utilizing CIO, we optimize the handover performance of the two users with the poorest performance. First, we utilize domain expertise to eliminate the HOF and RLF for user #1 by altering the CIO values between the serving and surrounding cells that are responsible for the handover issues. Results indicate that with CIO tuned for user #1, we are able to prevent handover failure and RLF for user #1. However, we can also see that unwanted handovers occur for static users as a result of CIO optimized for user #1. In addition, we observed a decline in the performance of other mobile users. For example, we notice handover failure and RLF instances for user #2 that are not existent with the default configuration. In the meantime, if we optimize the network to improve the performance of user #4, we observe that it

is possible to do so but at the expense of unwanted handovers of the static users or performance degradation for other users. These unwanted HO of static users and adverse impact to other mobile users can be avoided with UIO since we can target specific users instead of specific cells (i.e., assign UIO to mobile users and not to the static users).

6.2.2 UIO Compliance to 3GPP Standard Handover

The proposed UIO is designed to seamlessly integrate with the current 3GPP handover mechanism without compromising its cardinality [8]. As shown in Fig. 6.3, following the establishment of the Radio Resource Control (RRC), the source cell will acquire user attributes, including speed, trajectory, serving and neighboring cell IDs, and signal conditions. Leveraging these attributes, the UIO generator will calculate the UIO values using various methods, which will be discussed in the subsequent section. The serving base station attaches these UIO values to a measurement control message that includes event triggering criteria, such as offset, TTT, and CIO values.

Once the UE receives the measurement control message, it periodically measures parameters such as RSRP and RSRQ to track changes in signal conditions. If the specified event condition is met (eq. 4), the UE sends a measurement report (M) to the serving base station, which includes RSRP and RSRQ levels of the serving and target BSs. The serving base station uses this information to determine whether handover is necessary. If so, it sends a request for handover to the target base station, which responds by sending a handover command to the UE. The UE then detaches from the serving base station and initiates a random access channel (RACH) procedure to connect to the target cell.

The proposed UIO parameter can be easily adapted to current 5G networks, with minimal impact on the existing handover standards. The simplicity of the UIO

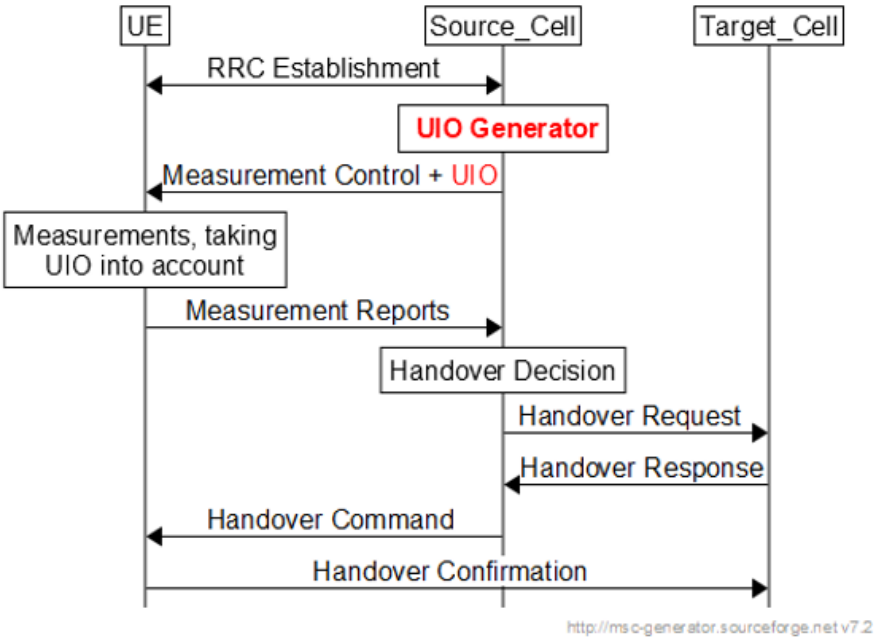


Fig. 6.3: HO message flow with the incorporation of the proposed UIO.

approach and its compatibility with current standards make it a promising solution for improving handover performance in wireless networks. In the next section, we discuss the different methods that can be used to calculate the values of UIO.

6.2.3 UIO Optimization Methods

Although UIO can enhance the handover performance of the network by introducing an additional degree of freedom, it simultaneously increases the complexity of determining its optimal value. Persistent handover problems can result from incorrectly setting the UIO values. In this section, we will examine several techniques for computing and optimizing the UIO values.

- Domain knowledge based manual setting: This technique involves identifying the underlying causes of handover issues, such as too early handover, handover to the wrong cell, too late handover, or ping-pong handover. This process requires a high level of domain expertise and practical experience to accurately determine the root cause of the problem. Once the underlying cause has been

identified, an appropriate UIO value can be selected to address the issue. For instance, a positive UIO value can be used to address delayed handovers, while negative UIO values can prevent premature handovers by neighboring cells.

However, this technique relies primarily on a hit-and-trial approach to determine the optimal UIO value for each user. Additionally, since the UIO values must be configured per user, scalability can be challenging given the sheer number of users. Moreover, adjusting the UIO values using this method may only produce a local improvement that is limited to a single user, rather than providing a comprehensive solution that addresses the handover issue across the entire network.

- Incorporation into SON solutions (i.e., MRO): MRO, or Mobility Robustness Optimization, represents an automated approach to adjusting UIO values based on user feedback. When users report radio link failures caused by different handover issues, such as late or early handover or handover to the wrong cell, the network analyzes the reports and determines the optimal UIO for each user using heuristics. While MRO offers a more automated solution than manual UIO adjustment based on domain knowledge, its effectiveness is dependent on the volume of statistical data available. Furthermore, the randomness of the ever-changing transmission environment and user behavior, such as mobility patterns and speed, can limit the effectiveness of UIO adjustments based on feedback. Finally, relying on this approach can make the UIO assignment reactive, which may not be suitable for emerging cellular networks.

- AI/ML with heuristic optimization (i.e., Genetic algorithm, Simulated Annealing) approach: At a more advanced level, RAN intelligence can observe and learn from a large number of handover events with associated parameters, and use this information to train AI/ML models to identify sets of UIO values

that result in successful handovers. This approach offers a more automated and scalable solution to the UIO optimization problem. Heuristic algorithms such as genetic algorithm and simulated annealing can be employed to search for the best combinations of the UIO that yield the maximum performance. This approach has the potential to improve handover performance significantly and reduce the need for manual intervention. However, training an AI/ML model with sparse training data can pose challenges to the utility of this approach. The effectiveness of the model depends on the quality and quantity of the training data. Therefore, there is a need for a large and diverse set of handover events and associated parameters to ensure that the AI/ML models are trained effectively. Despite these challenges, this approach holds great promise for the future of cellular networks, as it can lead to more efficient and effective handover management.

- Using Reinforcement Learning: The RL model aims to identify the optimal UIO for each user, maximizing KPIs like throughput, SINR, handover success rate, and load balancing. Using RL for UIO calculation has advantages over data-driven approaches like traditional AI/ML, as RL can learn without massive amounts of data. Instead, it learns iteratively through trial and error, based on the interaction of its agent and environment. By using RL-based UIO, network management and optimization tasks can be automated, leading to improved system performance and a better user experience with minimal human intervention.

- Graph Neural Network (GNN)-based UIO calculation: Graph Neural Networks (GNNs) are a type of artificial neural network designed to analyze and process complex relationships between entities in graph structured data. By using GNNs in combination with Reinforcement Learning (RL), a hybrid model can be created where GNNs are used to model the state of a graph-

structured environment, and RL is used to learn the best actions to take within that environment. For example, a graph could represent the state of a system, with nodes representing the current CIO values, user speed, the current serving cell, and the predicted cell for handover (which is determined by a mobility prediction model). Using this graph, users acting as RL agents can generate the optimal UIO values for the system. This hybrid approach enables the analysis of complex relationships between entities, improving the accuracy and efficiency of decision-making processes.

- Leveraging Digital Twin: Reinforcement Learning is a trial and error based learning process, which presents challenges for direct application in a live network due to potential network disruptions. To address this issue, we propose using a digital twin (DT) of the cellular network as a training ground for the RL model. A DT is a virtual representation of a physical system that is updated in real-time. Using a DT for RL training improves its utility, as the actual network is not affected during the training process. While 3GPP Rel. 17 (TR 37.817) [165] recommends AI/ML model training and inference be located at the Operations and Management layer (OAM) or at the base station itself, this approach poses risks, particularly for functions such as handover where the RL model must explore different UIO settings. To minimize the impact of the training process on the cellular network, we suggest using a digital twin to perform the exploration/exploitation task outside of the live network. This enables efficient training of the RL model while maintaining network stability.

6.3 Framework and System Model

In this section, we present the framework for calculating UIO values, and the KPIs that we aim to maximize, including SINR, HOSR, and signaling overhead. Next, we

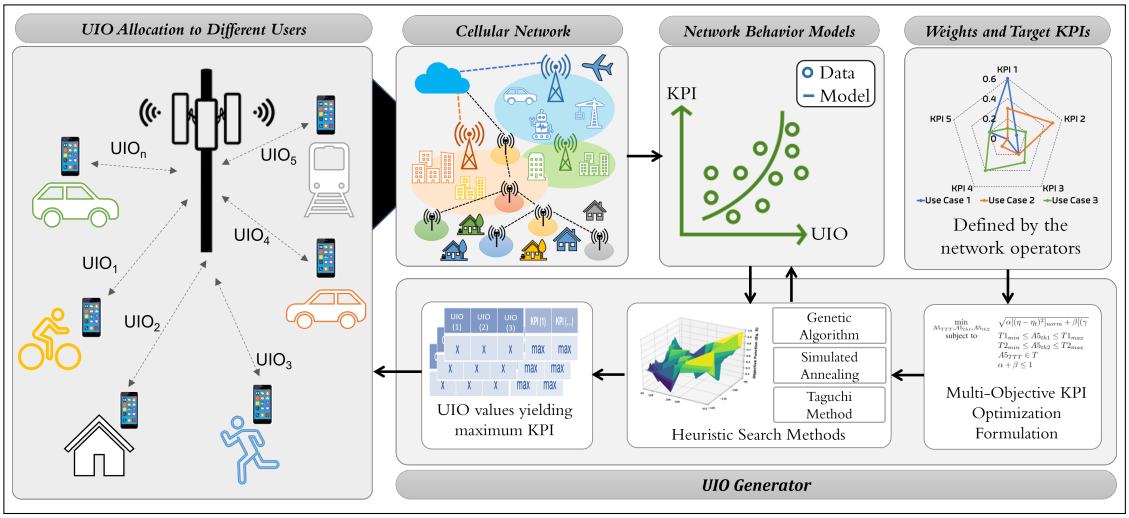


Fig. 6.4: Proposed UIO generation using ML with heuristics approach.

outline the COP-KPI optimization problem and provide details on the simulation setup.

6.3.1 Network-Originating UIO Generation Framework

The framework for network originating UIO, illustrated in Figure 6.4, comprises several steps. Firstly, machine learning models are employed to model network behaviors, i.e., the variations in Key Performance Indicators (KPIs). These models capture the relationship between the Configuration Parameters (COPs) and KPIs, with regression models being a common approach for estimating KPI values based on different sets of UIOs presented in tabular form. Hyperparameter optimization can be performed to achieve optimal model performance.

Once a well-trained machine learning model is obtained, it is combined with an optimization engine to optimize the UIO-KPI relationship. Unlike existing SON approaches that rely on a hit-and-trial method, our framework uses the model to identify the optimal set of COPs that can generate the expected new KPI values. Multi-objective optimization techniques, such as Genetic Algorithm, Taguchi Method, or Simulated Annealing, are employed to estimate the new KPI values for

a given set of UIOs. To enhance the customizability of the framework, network operators can assign weights to the KPIs based on their priorities. By assigning higher weights to a particular KPI, the optimization process is prioritized towards that KPI.

The heuristic search method then generates the optimal UIO combination that is relayed to each base station. This combination is then allocated to the users using the method described in Section 6.2.2. Overall, our proposed framework offers a more efficient and effective method for optimizing network performance, while also providing flexibility to accommodate diverse user priorities.

6.3.2 Problem Formulation

The objective of this chapter is to demonstrate the effectiveness of the UIO in optimizing wireless network performance by focusing on three critical KPIs: SINR, handover success rate (HOSR), and handover signaling overhead (SigO). By optimizing these KPIs, we aim to enhance the overall quality of experience (QoE) for wireless network users. Specifically, optimizing SINR can enhance the reliability and stability of wireless connections. In addition to indicating the quality of the received signal, SINR also provides an estimate of the network capacity. Meanwhile, maximizing HOSR can minimize the risk of connection dropouts during handovers. Moreover, optimizing HO signaling can reduce the SigO associated with handovers, improving network efficiency and reducing the likelihood of signaling-related errors.

Signal to interference and noise ratio

We leverage Atoll, a ray tracing commercial planning tool, to generate the RSRP map [166]. The aster propagation model is deployed for our path loss model, which uses ray tracing propagation techniques to incorporate vertical and horizontal diffraction/reflection. We use a practical 3-D antenna model composed of

horizontal and vertical antenna patterns instead of a theoretical antenna pattern equation. The resulting RSRP values from Atoll for user u from the serving BS s at physical resource block (PRB) j is denoted by $\lambda^u_{s,j}$. Using the RSRP values from Atoll, the SINR for a user u on PRB j allocated by BS s, denoted as $\gamma^u_{s,j}$, can be expressed as follows:

$$\gamma^u_{s,j} = \frac{\lambda^u_{s,j}}{K + \sum\limits_{\forall i \in B_f} \lambda^u_{i,j}} \tag{6.5}$$

where K is the thermal noise, the set $\mathbb{B}_f$ contains all the interfering base stations using the same frequency band as the user u and $\lambda^u_{i,j}$ represents the RSRP for user u from an interfering BS i at PRB j.

The SINR γ^u_s for user u connected to BS s can be computed by averaging the SINR $\gamma^u_{s,j}$ for all PRBs allocated to the user. The resulting SINR γ^u_s can be expressed as:

$$\gamma^u_s = \frac{\sum\limits_{\forall j \in \mathbb{R}_u} \gamma^u_{s,j}}{|\mathbb{R}_u|} \tag{6.6}$$

where the set $\mathbb{R}_u$ contains all the PRBs allocated to the user u. The mean SINR γ of all the users in the network can be written as:

$$\gamma = \frac{\sum\limits_{\forall i \in \mathbb{U}} \gamma^i_s}{|\mathbb{U}|} \tag{6.7}$$

where the set $\mathbb{U}$ contains all the users in the network.

Handover Success Rate

The provision of ultra-reliable low-latency communication (URLLC) is a critical requirement in the emerging mobile networks, particularly for applications such as intelligent transport systems and autonomous cars. Achieving this objective necessitates the optimization of the handover success rate (HOSR), which is a crucial key performance indicator that captures the effectiveness of handover-related parameter

settings. A poor HOSR could create a significant bottleneck for URLLC, leading to degraded user experience and potential safety hazards. HOSR ξ can be expressed as follows:

$$\xi = \frac{HOS}{HOS + HOF} \times 100\% \tag{6.8}$$

where HOS and HOF are the number of successful and failed handovers, respectively, in the network.

$$\begin{aligned} \max_{UIO} \quad & (\alpha[\gamma]_{\text{norm}}) + (\beta[\xi]_{\text{norm}}) + (1-\alpha-\beta)[1-\eta]_{\text{norm}}; \\ \text{subject to} \quad & UIO_{\min} \leq UIO \leq UIO_{\max} \\ & \alpha + \beta \leq 1 \end{aligned} \tag{6.9}$$

Signalling Overhead

Handovers are an important aspect of cellular networks, but they can also result in added SigO that can impact network performance. Although optimizing KPIs such as the number of HOF, handover latency, and SINR can enhance user QoE, they do not directly represent the burden caused by handovers on the network. To address this, we incorporate SigO as a KPI in our evaluation of handover performance.

To model the overhead of over-the-air signaling messages during handovers, we use the X2 interface [167]. The X2 interface enables communication between the source and target BS during handovers, and several types of signaling messages are transmitted over this interface. During the HO preparation phase, examples of signaling messages include measurement reports and handover commands (RRC connection reconfiguration). During the HO execution phase, synchronization, uplink allocation, and handover confirmation (RRC connection reconfiguration complete) messages are transmitted. In the event of handover failures, retransmission of these signaling messages can further increase SigO on the network.

To quantify the total handover SigO, we define the total additional signaling bytes transmitted over-the-air during the handover process as η. By taking into account both user QoE and SigO, we can better evaluate the overall performance of our system and identify potential areas for optimization.

Multi-objective Optimization Function

To jointly optimize of mean SINR, HOSR, and SigO, we formulate a multi-objective optimization problem that minimizes the differences of γ, ξ, and η with the target values of respective KPI. The optimization problem can be mathematically formulated as shown in eq. (6.9). The operator-defined weights, α and β, can be used to adjust the relative importance of SINR, HOSR, and SigO. We also normalize each KPI to remove bias towards larger KPI values. This approach ensures that the importance of each KPI is solely defined by their respective weights. The first constraint in eq. (6.9) limit the values of the optimization variables (i.e., UIO) to a pre-defined range. The second constraint states that the sum of the three weights is equal to one.

Solving the optimization problem posed in eq. (6.9) using an analytical approach is not feasible due to the difficulty, if not impossibility, of deriving tractable system-level models for SINR, HOSR, and SigO as a function of the UIO. Although some abstract mathematical models have been proposed [168, 169], they fail to capture the dynamics introduced by user mobility. To address this issue and enable practical self-optimization solutions, as originally proposed in [123], a more viable approach is to use data-driven modeling to solve (6.9).

6.3.3 Simulation Setup and Data Generation

Our simulation includes a network topology of 5G New Radio macro and small cells represented in Fig. 6.5. To improve the simulation environment's practicality,

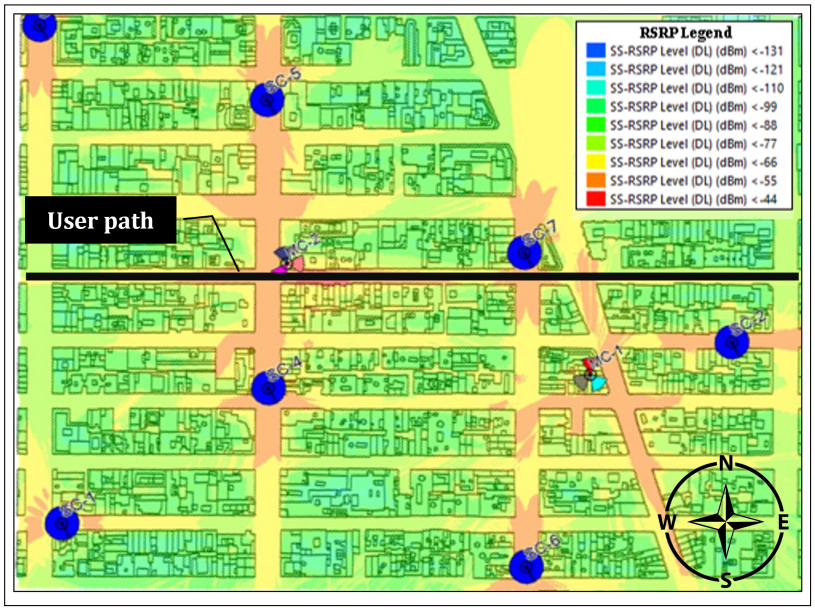


Fig. 6.5: Network deployment and RSRP traces.

we used Digital Terrain Model (DTM) raster files that contain the altitude value (in meters) of the ground, as well as features such as rivers and ridges. We also used clutter classes to represent different terrain types, including surface street, open bare ground, grassland, low vegetation, forest, and buildings of various height ranges. Clutter heights maps were used to represent individual heights of clutter. We incorporated the shadowing effect by using a log-normal (Gaussian) distribution, varying the standard deviation according to clutter type.

To conduct mobility simulations, we relied on SyntheticNET, a cutting-edge system-level simulator that complies with 3GPP standards [170]. Unlike Atoll, SyntheticNET can model 5G mobility parameters in detail, which makes it the ideal tool for our study. To import realistic network coverage data, we used Atoll and coupled it with SyntheticNET. We initialized the mobility parameters based on the GS setting of one of the leading operators in the USA and employed event A3 with an A3-offset of 2dB, TTT of 128ms, and hysteresis of 0dB to trigger the HO process.

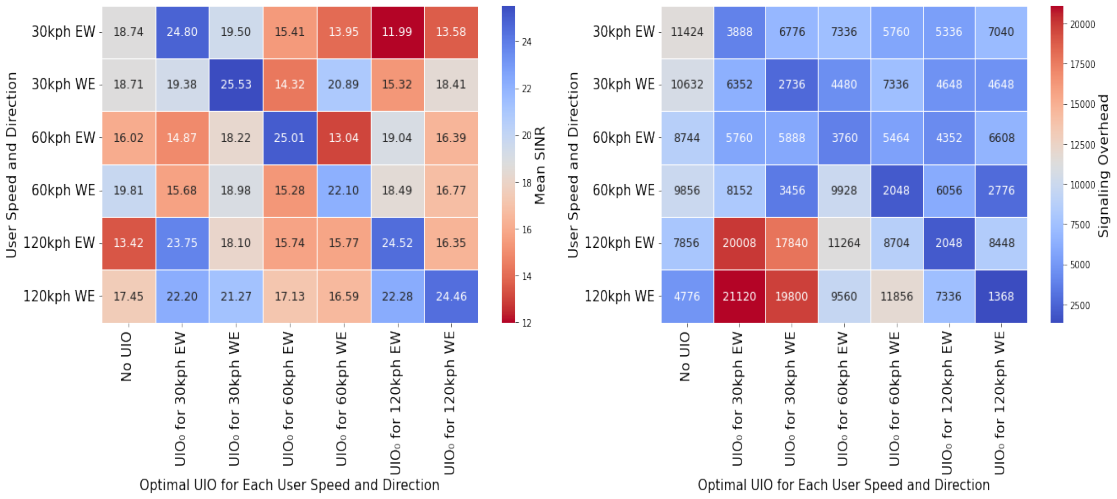
During the simulation, we deployed both mobile and static users within the network and ran it until the mobile users completed their path. As shown in Figure 3, the

Table 6.1: Description of Simulation Parameters

System parameters	Values
Carrier frequency	Small cell: 3.5 GHz
	Macro cell: 800 MHz
Channel bandwidth	Small cell: 50 MHz
	Macro cell: 5 MHz
Maximum transmit power	Small cell: 10 dBm
	Macro cell: 36 dBm
Path loss model	Aster propagation (ray tracing)
Cell sectors	Small: Omni-directional
	Macro: Tri-sectored
Number of sites	Macro:2
	Small: 7
Geographical information	Ground, building heights, land use map
Mobile users speed	30 km/h, 60 km/h, 120 km/h
Mobile users direction	East to West, West to East
UIO values	[-4, 2, 0, 2, 4] dB
Number of static users	60
Bin/grid size	1 m
Coverage area size	670000 m^2 (0.67 km^2)
BS height	Small cell: 10 m
	Macro cell: 30 m
Shadowing	Clutter-dependent shadowing
Sampling frequency	16 ms
Transmission time interval	1 ms
Total simulation time	108000 ms
Handover event	Event A3
A3-offset	2 dB
A3-TTT	128 ms
T310	1000 ms

mobile users' speeds varied between 30kph, 60kph, and 120kph, and their trajectory was from east (E) to west (W) and vice versa. To simulate network load and investigate the impact of CIO and UIO variations in static users, we also deployed 60 static users. For a comprehensive summary of our simulation parameters, please refer to Table 6.1.

The simulation cycle begins when the user moves from one end of the simulation area and ends when the mobile user reaches the edge of the area. During each cycle, the users are assigned different combinations of UIO values, which in this study range



(a) (b)

Fig. 6.6: Heatmap illustrating the influence of a user's optimal UIO (UIO_o) on other users' SINR (a) and SigO (b)

from [-4, -2, 0, 2, 4] dB. Each user is assigned UIO values based on their neighboring cells. If we follow the current standards, that would allow up to eight neighboring cells as potential targets for handover when a user is served by one base station. However, assigning eight different UIO values per user per neighboring cell would create an explosion of possible combinations, making data generation too complex. To reduce the complexity, we only consider the base stations that are serving along the tracks of the users. We also limit the number of neighboring cells to two or three for each base station, with the exception of one base station with five close neighbors. For instance, if a user is being served by BS 1 with three neighboring cells, we provide three UIO values for that user, one for each neighbor relation. With this approach, we generate around 18,750 data points consisting of different UIO combinations, user speed, and user direction. These data points are used to train our machine learning models and obtain the resulting KPIs.

6.4 Impact of UIO on KPIs

6.4.1 Impact of UIO in SINR, HOSR, and SigO

Fig. 6.6 illustrates the impact of different UIO values on both SINR and signalling overhead of different users. We compare the performance of the cases considering no UIO value with the case when UIO is optimized for each user denoted by UIO_O. This figure also illustrates the impact if the same value UIO_O for a specific user is used for other users, which happens in the case of CIO applied to all users. Fig. 6.6 (a) shows the variation in mean SINR with different UIO values. It can be observed that the optimal UIO combination for UEs moving at 30kph from east to west can produce SINR close to 25dB, which is almost 7dB higher than the scenario where no UIO is used. However, if this optimal UIO value is employed for other UEs with different speeds or directions, the achievable SINR is lower than the optimal. For instance, the UIO combination that is optimal for a user moving at 30kph east to west can only achieve a mean SINR of 15dB when used for a user moving at 60kph east to west, which is even lower than the scenario with no UIO. Nevertheless, the optimal UIO can be tailored for this particular user moving at 60kph east to west and the results indicate that this user can also achieve around 25dB mean SINR. This trend is observed in other users as well, indicating that a UIO combination that is optimal for one user may not be optimal for others. This further highlights the drawback of optimizing CIO, which can only be set to one value per neighbor for all the users of a BS. SINR can be better optimized if the offset is calibrated for each user separately enabled by the UIO proposed in this chapter.

Similarly, Fig. 6.6 (b) shows the same trend for signalling overhead. The results indicate that UIO offers a level of flexibility that CIO cannot match, as CIO lacks the capability to be uniquely assigned to each user, which in turn prevents the achievement of optimal KPIs for each user.

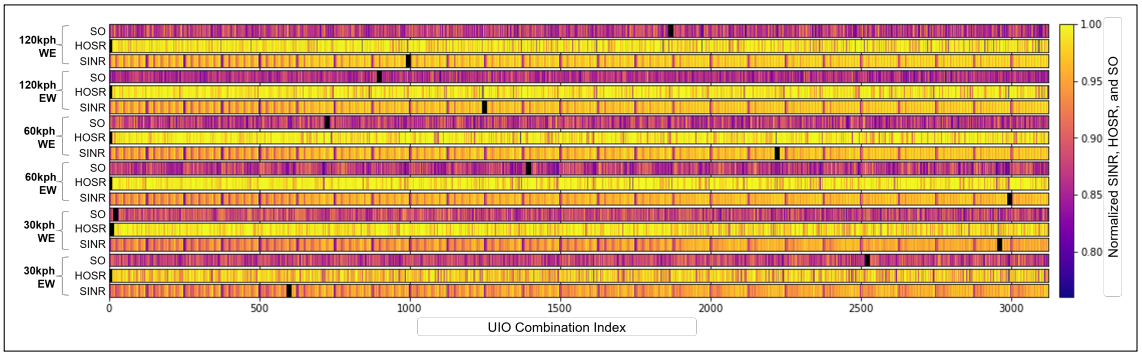


Fig. 6.7: Variations on the optimal UIO settings for different KPIs and user behavior.

6.4.2 Multi-objective Optimization

Fig. 6.7 presents a heat map that showcases all the three KPIs with varying UIO values for users with different speeds and directions. The x-axis represents the index of the UIO combinations based on the generated data. The black line in each row represents the optimal KPI value for each user. It is observed that the optimal black region for each KPI is different even for the same user. This indicates a tradeoff in optimizing the KPIs and hence the need for multi-objective optimization, as proposed in this chapter. In addition, the optimal value for each KPI changes with different user speed or direction of movement, which again signifies the importance of optimizing individual offsets for each user.

HOSR exhibits multiple optimal points for different combinations of data. On the other hand, unique COP combinations result in optimal SINR and SigO. For instance, a user moving at 30kph from east to west can achieve the maximum SINR using UIO combinations with an index of 607, while the maximum SigO can be attained using UIO combinations with an index of 2511. However, for a user with similar speed but moving in the opposite direction, the maximum SINR can be obtained using UIO combinations with an index of 2803, while the optimal SigO can be found using UIO combinations with a UIO combination index of 20. This shows that the user direction in addition to the user speed impact the optimal UIO combination.

6.5 ML-based with Hueristics UIO Generation

This section assess the effectiveness of machine learning models in conjunction with various heuristic approaches in identifying the ideal values of the UIO that maximize the objective function.

6.5.1 Performance Comparison of Machine Learning Models

This sub-section evaluates the performance of various machine learning algorithms in predicting KPIs using different combinations of UIO values. As discussed in the section 6.4, the user speed and direction also impact the best UIO combination that optimizes the KPI. Hence, we also provide user speed and direction to the machine learning along with the UIO combinations. The objective is to compare the performance of leading regression techniques. Tree based algorithms usually have better performance in the mobility management problems [171]. Hence, we compare the performance of decision tree, random forest, XGBoost, and LightGBM to identify the most effective algorithm for predicting the KPIs.

To achieve this goal, we utilize an 80%-20% train-test data split to assess the performance of the algorithms. Table 6.2 presents the resulting root mean square error (RMSE) for each algorithm on test data. Our findings indicate that the decision tree algorithm has the poorest performance, with RMSE values of 3.205 dB, 12.2%, and 2195.936 bytes for SINR, HOSR, and SigO, respectively. Among the tree-based algorithms, XGBoost and LightGBM performs the best. More specifically, LightGBM performs best in estimating the mean SINR and HOSR with RMSE of 2.151 dB and 7.8% respectively. Meanwhile, XGBoost performs best in predicting the mean SigO with an RMSE of 1580.239 bytes. However, the performance of LightGBM in estimating the SigO is also not far behind XGBoost with RMSE of 1591.896 bytes.

This chapter provides valuable insights into the effectiveness of machine learning

Table 6.2: Normalized RMSE comparison of different machine learning algorithms for SINR, HOSR, and SigO prediction.

KPI	XGBoost	LightGBM	Decision Tree Regressor	Random Forest
Mean SINR (dB)	2.238	**2.151**	3.205	2.423
HOSR (%)	8.0	**7.8**	11.2	8.7
Mean SigO (bytes)	**1580.239**	1591.896	2195.936	1743.766

algorithms for predicting KPIs using different UIO values. Our results suggest that tree-based algorithms, particularly XGBoost and LightGBM, show promise in accurately predicting KPIs, while decision tree algorithms are not well-suited for this task. These findings can be highly beneficial for network operators and researchers in developing and implementing machine learning algorithms for KPI prediction. By utilizing tree-based algorithms like XGBoost and LightGBM, network operators can optimize their networks and improve their overall performance. Researchers can also leverage our insights to advance the development of KPI prediction models for various applications, including network optimization and management.

6.5.2 Objective Function Optimization

Figure 6.8 displays the performance of the proposed machine learning (ML) model with heuristics in maximizing the key performance indicators (KPIs). When α and β are equal to 0.33, indicating equal priority is given to all three KPIs, LightGBM with genetic algorithm (GA) achieves a utility function value between 0.8 to 0.86 for the users. In contrast, LightGBM with simulated annealing (SA) demonstrates a utility function value of approximately 0.78 to 0.85. While brute force yields higher utility function values, it requires a large number of iterations to determine the optimal values. For instance, in our case, the potential combinations of UIO are approximately 5^{19}. This value escalates with the increase in the number of UIO ranges and the number of base stations. On the other hand, GA and SA only takes 1000 iterations to converge at near-optimal values.

Furthermore, it can be observed that the difference between the optimal value re-

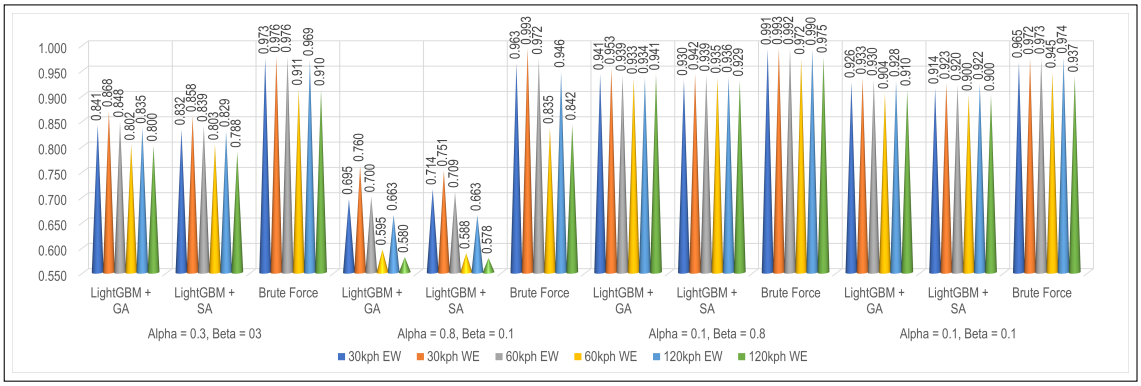


Fig. 6.8: Performance comparison between machine learning with heuristic optimization approach and brute force methods, while also examining the impact of varying the weights.

turned by the heuristic approaches and brute force changes with varying the KPI weights. For example, when α and β are set to 0.1 and 0.8, respectively, indicating a lower priority in optimizing the SINR, the utility function values become comparable to brute force (0.92 or above for all the users). A similar trend is observed when both α and β are set to 0.1 giving a high weight to optimizing SigO. However, it is worth noting that both GA and SA optimization methods exhibit a decrease in performance when more weight is given to α, indicating higher priority to SINR. This is due to the more unpredictable nature of SINR, which ML models may fail to capture as accurately as handover success rate and SigO. The highest difference between the brute force and both SA and GA is observed when α and β are set to 0.8 and 0.1, respectively. This strengthens the point that SINR is a difficult KPI to accurately capture for ML models.

To compare the performance of UIO with CIO, we conducted an experiment where we varied the CIO values and generated the corresponding data. Fig. 6.9 indicates that the objective function value is higher when using UIO as the parameter instead of CIO. This trend is observed for both GA and SA. By combining LightGBM with GA, we were able to achieve an objective function of 0.833 with UIO optimization, whereas the use of CIO only yielded a maximum of 0.813. The difference in performance is even more pronounced when SA is utilized. Specifically, UIO-optimized

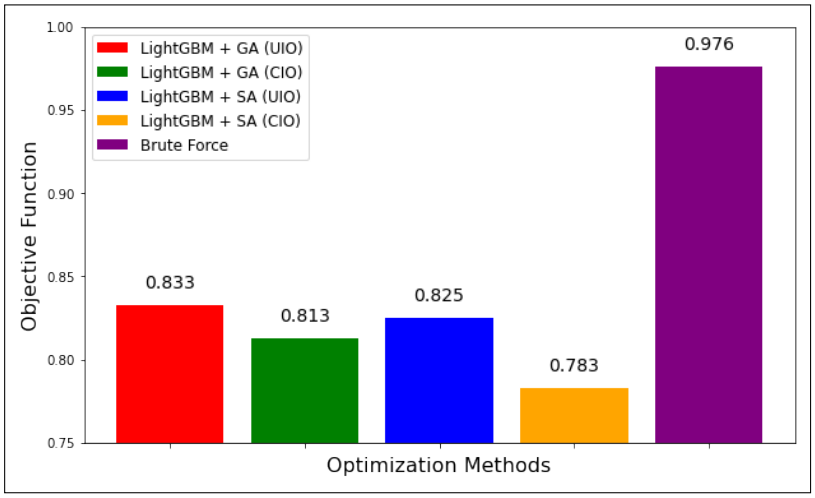


Fig. 6.9: Performance comparison between CIO-optimized and UIO-optimized networks.

networks display an objective function value of 0.825, whereas networks optimized using CIO exhibit a lower objective function value of 0.783. This indicates the superior performance of UIO compared to CIO.

6.6 Conclusion

This chapter introduces UIO, a novel parameter that addresses the limitations of the commonly used CIO parameter. UIO provides greater flexibility and granularity in controlling the handover behavior of individual users. Moreover, this new parameter is well-suited for current handover standards, making it highly adaptable. We present several methods for calculating the value of UIO, and our results show that leveraging machine learning with heuristics is a simple yet effective approach that outperforms CIO in optimizing multiple KPIs, including SINR, HOSR, and signaling overhead.

CHAPTER 7

Conclusions and Future Work

The cellular network industry has undergone significant evolution in recent years, and mobility management has become a major challenge. Emerging cellular networks have peculiarities such as densified deployment, stringent Quality of Service (QoS) requirements, and evolved network architecture, which make efficient mobility management crucial. However, several roadblocks hinder the development of a more robust, proactive, and autonomous mobility management scheme. This dissertation presents these roadblocks and proposes solutions to address them. One challenge is the lack of suitable tools to investigate mobility, which is caused by complexity and missing system level details. To tackle this challenge, a tri-pronged approach is presented, including the development of a simulator, the deployment of an experimental testbed called TurboRAN, and an AI-enabled handover parameter optimization framework. Another challenge is the increasing prevalence of radio link failures caused by handover failures. To address this issue, a novel data-driven solution called TORIS is proposed, which reduces inter-frequency handover failures by developing and integrating two sub-solutions: an AI-based model to predict inter-frequency HOFs and a heuristic scheme to tune the transmit power of serving and target cells. The third challenge pertains to the limitations of the existing handover parameter optimization standard, which demands tradeoffs between KPIs. To address this challenge, a novel parameter called user individual offset (UIO) is proposed, which considers various user behaviors for value selection and helps solve age-old challenges in mobility management without tradeoffs.

For future work, we plan to further investigate the potential of alternative methods for calculating UIO values including reinforcement learning-based, graph neural

networks-based and digital twin-based approaches. Additionally, we will explore the potential of user-determined UIO, which represents a more futuristic approach that may be applicable in 6G networks. We will also utilize TurboRAN to test its real-world impact. This user-centric approach has the potential to significantly enhance handover system performance and represents an exciting avenue for further research.

Bibliography

[1] J. Joseph, F. Ahmed, T. Jokela, O. Tirkkonen, J. Poutanen, and J. Niemela, "Big data enabled mobility robustness optimization for commercial LTE networks," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[2] W. Feng, Y. Wang, D. Lin, N. Ge, J. Lu, and S. Li, "When mmwave communications meet network densification: A scalable interference coordination perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1459–1471, 2017.

[3] M. Mollel, M. Ozturk, M. Kisangiri, S. Kaijage, O. Onireti, M. A. Imran, and Q. H. Abbasi, "Handover management in dense networks with coverage prediction from sparse networks," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, 2019, pp. 1–6.

[4] B. Zhang, W. Qi, and J. Zhang, "An energy efficiency and ping-pong handover ratio optimization in two-tier heterogeneous networks," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 532–536.

[5] M. U. B. Farooq, M. Manalastas, S. M. A. Zaidi, A. Abu-Dayya, and A. Imran, "Machine learning aided holistic handover optimization for emerging networks," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 710–715.

[6] D. V. Zavyalova, M. L. Rolich, and A. V. Andreev, "Definition the optimal parameters of handover procedures in lte networks," in *2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, 2016, pp. 110–113.

[7] M. Manalastas, M. U. B. Farooq, S. M. A. Zaidi, A. Abu-Dayya, and A. Imran, "A data-driven framework for inter-frequency handover failure prediction and mitigation," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6158–6172, 2022.

[8] "3gpp ts 36.331: "evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC) protocol specification"."

[9] J. Joseph, F. Ahmed, T. Jokela, O. Tirkkonen, J. Poutanen, and J. Niemela, "Big data enabled mobility robustness optimization for commercial lte networks," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[10] D. M. Rose, J. Baumgarten, S. Hahn, and T. Kurner, "Simone - simulator for mobile networks: System-level simulations in the context of realistic scenarios," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–7.

[11] C. Jao, C. Wang, T. Yeh, C. Tsai, L. Lo, J. Chen, W. Pao, and W. Sheen, "Wise: A system-level simulator for 5g mobile networks," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 4–7, 2018.

[12] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6g networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, 2020.

[13] C. Benzaid and T. Taleb, "Ai-driven zero touch network and service management in 5g and beyond: Challenges and research directions," *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.

[14] S.-C. Lin, C.-H. Lin, and W.-C. Chen, "Zero-touch network on industrial iot: An end-to-end machine learning approach," 2022.

[15] P. Muñoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, 2013.

[16] D. Castro-Hernandez and R. Paranjape, "Optimization of handover parameters for LTE/LTE-A in-building systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5260–5273, 2017.

[17] X. Cai, C. Wu, J. Sheng, J. Zhang, and Y. Wang, "A parameter optimization method for LTE-R handover based on reinforcement learning," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 1216–1221.

[18] A. S. Priyadharshini and P. T. V. Bhuvaneswari, "A study on handover parameter optimization in LTE-A networks," in *2016 International Conference*

on Microelectronics, Computing and Communications (MicroCom), 2016, pp. 1–5.

[19] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for LTE systems," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011, pp. 1647–1651.

[20] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved QoE in LTE," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 794–798.

[21] J. Xu, Y. Zhao, and X. Zhu, "Mobility model based handover algorithm in LTE-Advanced," in *2014 10th International Conference on Natural Computation (ICNC)*, 2014, pp. 230–234.

[22] K. S. Lee and A. Park, "Reduction of handover failure for small cells in heterogeneous networks," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, 2014, pp. 707–708.

[23] M. B. Gharbia and R. Bouallegue, "Handover decision algorithm in femtocell long term evolution networks," in *2018 Seventh International Conference on Communications and Networking (ComNet)*, 2018, pp. 1–6.

[24] J. Shodamola, U. Masood, M. Manalastas, and A. Imran, "A machine learning based framework for kpi maximization in emerging networks using mobility parameters," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020, pp. 1–6.

[25] M. Z. Asghari, M. Ozturk, and J. Hämäläinen, "Reinforcement learning based mobility load balancing with the cell individual offset," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.

[26] A. Madelkhanova, Z. Becvar, and T. Spyropoulos, "Optimization of cell individual offset for handover of flying base stations and users," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.

[27] G. Alsuhli, H. A. Ismail, K. Alansary, M. Rumman, M. Mohamed, and K. G. Seddik, "Deep reinforcement learning-based cio and energy control for lte mo-

bility load balancing," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[28] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6g: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[29] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5g and beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

[30] Haneya Naeem Qureshi, Usama Masood , Syed Muhammad Asad Zaidi, Hasan Farooq, Maxime Bouton, Shruti Bothe, Per Karlsson, Ali Rizwan, and A. Imran, "Towards Addressing Training Data Scarcity Challenge in Emerging Radio Access Networks: A Survey and Framework (Submitted)," *IEEE Communication Surveys and Tutorials*, 2022.

[31] N. I. Sarkar and S. A. Halim, "A review of simulation of telecommunication networks: simulators, classification, comparison, methodologies, and recommendations," 2011.

[32] A. S. Toor and A. K. Jain, "A survey on wireless network simulators," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 1, pp. 62–69, 2017. [Online]. Available: https://beei.org/index.php/EEI/article/view/568

[33] M. H. Kabir, S. Islam, M. J. Hossain, and S. Hossain, "Detail comparison of network simulators," *International Journal of Scientific & Engineering Research*, vol. 5, no. 10, pp. 203–218, 2014.

[34] S. Siraj, M. A. K. Gupta, and M. Rinku-Badgujar, "Network simulation tools survey," 2012.

[35] L. Ronit, J. Maharshi, and J. Amit, "Survey on network simulators," *International Journal of Computer Applications*, vol. 182, pp. 23–30, 2018.

[36] S. G. Gupta, M. M. Ghonge, P. D. Thakare, and P. Jawandhiya, "Open-source network simulation tools: An overview," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, pp. 1629–1635, 2013.

[37] A. Kumar, S. K. Kaushik, R. Sharma, and P. Raj, "Simulators for wireless networks: A comparative study," in *2012 International Conference on Computing Sciences*, 2012, pp. 338–342.

[38] L. Hogie, P. Bouvry, and F. Guinand, "An overview of manets simulation," *Electronic notes in theoretical computer science*, vol. 150, no. 1, pp. 81–101, 2006.

[39] I. Dorathy and M. Chandrasekaran, "Simulation tools for mobile ad hoc networks: a survey," *Journal of applied research and technology*, vol. 16, pp. 437 – 445, 00 2018. [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232018000500437&nrm=iso

[40] Manpreet and J. Malhotra, "A survey on manet simulation tools," in *2014 Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH)*, 2014, pp. 495–498.

[41] G. Chengetanai and G. B. O'Reilly, "Survey on simulation tools for wireless mobile ad hoc networks," in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2015, pp. 1–7.

[42] M. N. Jambli, H. Lenando, K. Zen, S. M. Suhaili, and A. Tully, "Simulation tools for mobile ad-hoc sensor networks: A state-of-the-art survey," in *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*. IEEE, 2012, pp. 1–6.

[43] N. Bhatt, "Comarison and analysis of simulators for ad hoc wireless networks," *IOSR Journal of Engineering*, vol. 3, pp. 14–22, 2013.

[44] E. Spaho, L. Barolli, G. Mino, F. Xhafa, and V. Kolici, "Vanet simulators: A survey on mobility and routing protocols," in *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*. IEEE, 2011, pp. 1–10.

[45] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (vanets)," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813–828, 2011.

[46] P. Owczarek and P. Zwierzykowski, "Review of simulators for wireless mesh

networks," *Journal of Telecommunications and Information technology*, 2014.

[47] M. Imran, A. M. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *2010 International Symposium on Information Technology*, vol. 2. IEEE, 2010, pp. 897–902.

[48] M. Korkalainen, M. Sallinen, N. K¨arkk¨ainen, and P. Tukeva, "Survey of wireless sensor networks simulation tools for demanding applications," in *2009 Fifth international conference on networking and services*. IEEE, 2009, pp. 102–106.

[49] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative study of wireless network simulators," in *Seventh International Conference on Networking (icn 2008)*. IEEE, 2008, pp. 517–523.

[50] C. P. Singh, O. Vyas, and M. K. Tiwari, "A survey of simulation in sensor networks," in *2008 International Conference on Computational Intelligence for Modelling Control & Automation*. IEEE, 2008, pp. 867–872.

[51] B. Musznicki and P. Zwierzykowski, "Survey of simulators for wireless sensor networks," *International Journal of Grid and Distributed Computing*, vol. 5, no. 3, pp. 23–50, 2012.

[52] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless sensor network simulators a survey and comparisons," *International Journal of Computer Networks*, vol. 2, no. 5, pp. 249–265, 2011.

[53] M. Jevtić, N. Zogović, and G. Dimić, "Evaluation of wireless sensor network simulators," in *Proceedings of the 17th telecommunications forum (TELFOR 2009), Belgrade, Serbia*. Citeseer, 2009, pp. 1303–1306.

[54] F. Yu and R. Jain, "A survey of wireless sensor network simulation tools," *Washington University in St. Louis, Department of Science and Engineering*, 2011.

[55] S. Yadav and A. Chitra, "Wireless sensor networks-architectures, protocols, simulators and applications: a survey," *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 4, pp. 1941–1953, 2012.

[56] M. A. Khan, H. Hasbullah, and B. Nazir, "Recent open source wireless sensor network supporting simulators: A performance comparison," in *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, 2014, pp. 324–328.

[57] A. Stetsko, M. Stehlik, and V. Matyas, "Calibrating and comparing simulators for wireless sensor networks," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 733–738.

[58] C. Bouras, G. Diles, A. Gkamas, and A. Zacharopoulos, "A comparative study of 4g and 5g network simulators," *International Journal on Advances in Networks and Services, IARIA*, vol. 13, 2020.

[59] P. K. Gkonis, P. T. Trakadas, and D. I. Kaklamani, "A comprehensive study on simulation techniques for 5g networks: State of the art results, analysis, and future challenges," *Electronics*, vol. 9, no. 3, p. 468, Mar 2020. [Online]. Available: http://dx.doi.org/10.3390/electronics9030468

[60] G. Kecskemeti, G. Casale, D. N. Jha, J. Lyon, and R. Ranjan, "Modelling and simulation challenges in internet of things," *IEEE Cloud Computing*, vol. 4, no. 1, pp. 62–69, 2017.

[61] Y. Wang, J. Xu, and L. Jiang, "Challenges of system-level simulations and performance evaluation for 5g wireless networks," *IEEE Access*, vol. 2, pp. 1553–1561, 2014.

[62] S. Cho, S. Chae, M. Rim, and C. G. Kang, "System level simulation for 5g cellular communication systems," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017, pp. 296–299.

[63] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "SyntheticNET: A 3GPP Compliant Simulator for AI Enabled 5G and Beyond," *IEEE Access*, vol. 8, pp. 82 938–82 950, 2020.

[64] S. Martiradonna, A. Grassi, G. Piro, and G. Boggia, "Understanding the 5g-air-simulator: A tutorial on design criteria, technical components, and reference use cases," *Computer Networks*, vol. 177, p. 107314, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128619317347

[65] R. I. Tinini, M. R. P. dos Santos, G. B. Figueiredo, and D. M. Batista, "5gpy: A simpy-based simulator for performance evaluations in 5g hybrid cloud-fog ran architectures," *Simulation Modelling Practice and Theory*, vol. 101, p. 102030, 2020, modeling and Simulation of Fog Computing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X19301613

[66] J. C. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of lte networks," in *2010 IEEE 71st Vehicular Technology Conference*, 2010, pp. 1–5.

[67] Y. Kim, J. Bae, J. Lim, E. Park, J. Baek, S. I. Han, C. Chu, and Y. Han, "5G K-Simulator: 5G System Simulator for Performance Evaluation," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2018, doi: 10.1109/DySPAN.2018.8610404.

[68] N. Mohsen and K. S. Hassan, "C-ran simulator: A tool for evaluating 5g cloud-based networks system-level performance," in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2015, pp. 302–309.

[69] P. Hegyi and J. Varga, "Telco cloud simulator," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–7.

[70] V. V. Diaz and D. Marcano Aviles, "A Path Loss Simulator for the 3GPP 5G Channel Models," in *IEEE International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2018.

[71] Q. Zhu, C.-X. Wang, B. Hua, K. Mao, S. Jiang, and M. Yao, "3gpp tr 38.901 channel model," *Wiley 5G Ref: The Essential 5G Reference Online*, pp. 1–35, 2019.

[72] G. Pereyra, C. Rattaro, and P. Belzarena, "Py5chesim: a 5g multi-slice cell capacity simulator," in *2021 XLVII Latin American Computing Conference (CLEI)*, 2021, pp. 1–8.

[73] M. Han, J. W. Lee, C. G. Kang, and M. J. Rim, "5g k-simsys: Open/modular/flexible system level simulator for 5g system," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2018, pp. 1–2.

[74] I. Belikaidis, A. Georgakopoulos, E. Kosmatos, I. de-la-Bandera, D. Palacios, R. Barco, and P. Demestichas, "5g component carrier management evaluation by means of system level simulations," in *2019 European Conference on Networks and Communications (EuCNC)*, 2019, pp. 592–596.

[75] K. Bąkowski, M. Rodziewicz, and P. Sroka, "System-level simulations of selected aspects of 5g cellular networks," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, 2015, pp. 711–715.

[76] M. Liu, P. Ren, Q. Du, W. Ou, X. Xiong, and G. Li, "Design of system-level simulation platform for 5g networks," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2016, pp. 1–6.

[77] M. K. Müller, F. Ademaj, T. Dittrich, A. Fastenbauer, B. R. Elbal, A. Nabavi, L. Nagel, S. Schwarz, and M. Rupp, "Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 17, Sep. 2018.

[78] X. Wang, Y. Chen, and Z. Mai, "A Novel Design of System Level Simulator for Heterogeneous Networks," in *IEEE Globecom Workshops (GC Wkshps)*, 2017.

[79] I. Viering, A. Lobinger, and S. Stefanski, "Efficient uplink modeling for dynamic system-level simulations of cellular and mobile networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–15, 2010.

[80] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[81] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A survey on green 6g network: Architecture and technologies," *IEEE Access*, vol. 7, pp. 175 758–175 768, 2019.

[82] F. Tariq, M. R. Khandaker, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A speculative study on 6g," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 118–125, 2020.

[83] I. F. Akyildiz, A. Kak, and S. Nie, "6g and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, 2020.

[84] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6g: A comprehensive survey," *IEEE Open Journal of the Communications Society*, 2021.

[85] G. Piro, N. Baldo, and M. Miozzo, "An lte module for the ns-3 network simulator," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '11. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, p. 415–422.

[86] "Atoll Overview ," Accessed on: March 16, 2022 [Online]. Available: https://www.forsk.com/atoll-overview.

[87] M. 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 turboran 5g and beyond testbed," *IEEE Access*, vol. 10, pp. 39 810–39 824, 2022.

[88] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.

[89] "5G Toolbox: Simulate, analyze, and test 5G communications systems," [Online]. Available: https://www.mathworks.com/products/5g.html.

[90] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmwave massive mimo through deep learning," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 325–10 329, 2019.

[91] M. Dardaillon, K. Marquet, T. Risset, and A. Scherrer, "Software defined radio architecture survey for cognitive testbeds," in *IEEE 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012, pp. 189–194.

[92] X. Lu, L. Ni, S. Jin, C.-K. Wen, and W.-J. Lu, "SDR implementation of a real-time testbed for future multi-antenna smartphone applications," *IEEE Access*, vol. 5, pp. 19 761–19 772, 2017.

[93] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, "Versatile Mobile Communications Simulation: The Vienna 5G Link Level Simulator," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018, doi: https://doi.org/10.1186/s13638-018-1239-6.

[94] M. Penner, M. Nabeel, and J. Peissig, "FBMC Testbed with Frequency Domain Synchronization and Adaptive Bandwidth," in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2021, pp. 423–426.

[95] K. Kondepu, F. Giannone, S. Vural, B. Riemer, P. Castoldi, and L. Valcarenghi, "Experimental demonstration of 5G virtual EPC recovery in federated testbeds," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 712–713.

[96] "Ericsson 5G radio test bed biggest contribution to 5G development in Asia ," Accessed on: March 16, 2022 [Online]. Available: https://www.ericsson.com/en/news/2015/10/ericsson-5g-radio-test-bed-biggest-contribution-to-5g-development-in-asia.

[97] "5TONIC: an open research and innovation laboratory focusing on 5G technologies ," Accessed on: March 16, 2022 [Online]. Available: https://www.5tonic.org.

[98] "5G Playground ," Accessed on: March 16, 2022 [Online]. Available: https://www.fokus.fraunhofer.de/go/en/fokus_testbeds/5g_playground.

[99] "NITOS - Network Implementation Testbed using Open Source platforms ," Accessed on: March 16, 2022 [Online]. Available: https://nitlab.inf.uth.gr/NITlab/.

[100] "SK Telecom opens 5G Playground to lead Innovation towards 5G Commercialization ," Accessed on: March 16, 2022 [Online]. Available: https://www.5tonic.org.

[101] "University of Bristol 5G Testbed ," Accessed on: March 16, 2022 [Online]. Available: https://5ginfire.eu/university-of-bristol-5g-testbed/.

[102] "ESPOO Aalto 5G Research Infrastructure ," Accessed on: March 16, 2022

[Online]. Available: http://5gtnf.fi/sites/espoo/.

[103] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The world's first real-time testbed for massive MIMO: Design, implementation, and validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.

[104] "TitanMIMO-6 Sub 6 GHz Massive MIMO Testbed Product Sheet ," Accessed on: March 16, 2022 [Online]. Available: https://docplayer.net/25673180-Titanmimo-6-sub-6-ghz-massive-mimo-testbed-product-sheet-nutaq-com-montreal-quebec.html.

[105] "5G/6G Innovation Centre Membership Categories," Accessed on: March 16, 2022 [Online]. Available: https://www.surrey.ac.uk/institute-communication-systems/5g-6G-innovation-centre/become-member.

[106] T. R. Newman, S. S. Hasan, D. DePoy, T. Bose, and J. H. Reed, "Designing and Deploying a Building-wide Cognitive Radio Network Testbed," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 106–112, 2010.

[107] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, and B. A. Floyd, "Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAW Architecture," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 22–30, 2020.

[108] A. Banerjee, J. Cho, E. Eide, J. Duerig, B. Nguyen, R. Ricci, J. Van der Merwe, K. Webb, and G. Wong, "Phantomnet: Research Infrastructure for Mobile Networking, Cloud Computing and Software-defined Networking," *GetMobile: Mobile Computing and Communications*, vol. 19, no. 2, pp. 28–33, 2015.

[109] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, D. Maas, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L. B. Stoller, J. Van der Merwe, K. Webb, and G. Wong, "POWDER: Platform for Open Wireless Data-Driven Experimental Research," in *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation Characterization*, 2020, p. 17–24. [Online]. Available: https://doi.org/10.1145/3411276.3412204

[110] "TurboRAN: State of the art testbed for enabling unprecedented experimental research on multi-band, multi-tier artificial intelligence enabled wire-

less networks of future ," Accessed on: March 16, 2022 [Online]. Available: https://ai4networks.com/turboran.php.

[111] Y. Hao, M. Chen, L. Hu, J. Song, M. Volk, and I. Humar, "Wireless fractal ultra-dense cellular networks," *Sensors*, vol. 17, no. 4, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/4/841

[112] "Radio Communication Test Station, MT8000A ," Accessed on: March 16, 2022 [Online]. Available: https://www.anritsu.com/en-us/test-measurement/products/mt8000a.

[113] "Challenge: Capturing the Promises of 5G-RAN ," Accessed on: March 16, 2022 [Online]. Available: https://www.keysight.com/us/en/assets/7019-0128/data-sheets/5G-RAN-Testing.pdf.

[114] "Amarisoft Callbox ," Accessed on: March 16, 2022 [Online]. Available: https://www.amarisoft.com/app/uploads/2021/05/AMARI-Callbox-Ultimate.pdf.

[115] "AMARI Callbox Series ," Accessed on: March 16, 2022 [Online]. Available: https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/.

[116] "Commscope VVSSP-65S-R1BV2 Antenna ," Accessed on: March 16, 2022 [Online]. Available: https://www.commscope.com/product-type/antennas/base-station-antennas-equipment/base-station-antennas/itemvvssp-65s-r1bv2/.

[117] "AW3639 Data Sheet ," Accessed on: March 16, 2022 [Online]. Available: https://alphawireless.com/ds/AW3639.pdf.

[118] "Comba Telecom to Supply Latin America's First Multi-vendor Antenna Optimization System ," Accessed on: March 16, 2022 [Online]. Available: https://www.comba-telecom.com/en/press-releases/industry-releases/item/643-comba-telecom-to-supply-latin-america-s-first-multi-vendor-antenna-optimization-system.

[119] "Remote Electrical Tilt (RET) Equipment ," Accessed on: March 16, 2022 [Online]. Available: https://www.commscope.com/product-type/antennas/base-station-antennas-equipment/remote-electrical-tilt-ret-equipment/.

[120] "CCU-1 Electrical Downtilt Control Unit ," Accessed on: March 16, 2022 [Online]. Available: https://www.tessco.com/product/ccu-1-electrical-downtilt-control-unit-295806.

[121] S.-J. Yoo, D. Cypher, and N. Golmie, "Timely effective handover mechanism in heterogeneous wireless networks," *Wireless Personal Communications*, vol. 52, no. 3, pp. 449–475, 2010.

[122] A. Asghar, H. Farooq, and A. Imran, "Self-healing in emerging cellular networks: Review, challenges, and research directions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1682–1709, 2018.

[123] 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.

[124] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[125] 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.

[126] M. U. Bin Farooq, M. Manalastas, W. Raza, A. Ijaz, S. M. A. Zaidi, A. Abu-Dayya, and A. Imran, "Data driven optimization of inter-frequency mobility parameters for emerging multi-band networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[127] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[128] B. Hughes, S. Bothe, H. Farooq, and A. Imran, "Generative adversarial learning for machine learning empowered self organizing 5g networks," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 282–286.

[129] C. Parera, A. E. Redondi, M. Cesana, Q. Liao, and I. Malanchini, "Transfer learning for channel quality prediction," in *2019 IEEE International Symposium on Measurements Networking (MN)*, 2019, pp. 1–6.

[130] 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.

[131] H. Farooq, J. Forgeat, S. Bothe, M. Bouton, M. Shirazipour, and P. Karlsson, "Coordinated hyper-parameter search for edge machine learning in beyond-5g networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–7.

[132] T. T. Le, W. Fu, and J. H. Moore, "Scaling tree-based automated machine learning to biomedical big data with a feature set selector," *Bioinformatics*, vol. 36, no. 1, pp. 250–256, 2020.

[133] M. Manalastas, "Realistic modeling of handover events in a multi-carrier 5G network: A preliminary step towards COP-KPI relationship realization," *Master's Thesis, University of Oklahoma-Tulsa*, 2020.

[134] E. U. T. R. Access, "Self-configuring and self-optimizing network use cases and solutions," *Protocol specification (Release 9)*, 2011.

[135] S. Khunteta and A. K. R. Chavva, "Deep learning based link failure mitigation," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 806–811.

[136] Xiaomeng Shu, Li Zhu, Hongli Zhao, and Tao Tang, "A novel handoff decision algorithm in TD-LTE based train-ground communication system," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 757–761.

[137] H. Qu, Y. Zhang, J. Zhao, G. Ren, and W. Wang, "A hybrid handover forecasting mechanism based on fuzzy forecasting model in cellular networks," *China Communications*, vol. 15, no. 6, pp. 84–97, 2018.

[138] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "Deepchannel: Wireless channel quality prediction using deep learning," *IEEE Transactions*

on Vehicular Technology, vol. 69, no. 1, pp. 443–456, 2020.

[139] "Fujitsu Whitepape: Enhancing LTE Cell-Edge Performance via PDCCH ICIC."

[140] Y. Mal, J. Chen, and H. Lin, "Mobility robustness optimization based on radio link failure prediction," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 454–457.

[141] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

[142] H. Wang, Z. Wu, S. Ma, S. Lu, H. Zhang, G. Ding, and S. Li, "Deep learning for signal demodulation in physical layer wireless communications: Prototype platform, open dataset, and analytics," *IEEE Access*, vol. 7, pp. 30 792–30 801, 2019.

[143] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. F. Kaijage, M. Kisangiri, S. Hussain, M. A. Imran, and Q. H. Abbasi, "A survey of machine learning applications to handover management in 5g and beyond," *IEEE Access*, vol. 9, pp. 45 770–45 802, 2021.

[144] M. Tayyab, G. P. Koudouridis, and X. Gelabert, "A simulation study on LTE handover and the impact of cell size," in *Broadband Communications, Networks, and Systems*, V. Sucasas, G. Mantas, and S. Althunibat, Eds. Cham: Springer International Publikoularidou,shing, 2019, pp. 398–408.

[145] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *CoRR*, vol. abs/1106.1813, 2011. [Online]. Available: http://arxiv.org/abs/1106.1813

[146] J. Young, P. Graham, and R. Penny, "Using bayesian networks to create synthetic data," *Journal of Official Statistics*, vol. 25, no. 4, p. 549, 2009.

[147] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[148] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2016, pp. 399–410.

[149] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Advances in Neural Information Processing Systems*, 2019.

[150] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887.

[151] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[152] J. Zhang and X. Li, "Phishing detection method based on borderline-smote deep belief network," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2017, pp. 45–53.

[153] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017.

[154] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, "Generation and evaluation of synthetic patient data," *BMC Medical Research Methodology*, vol. 20, pp. 1–40, 2020.

[155] I. Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and Computers in Simulation*, vol. 55, no. 1, pp. 271 – 280, 2001, the Second IMACS Seminar on Monte Carlo Methods. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378475400002706

[156] S. Chang, G. Li, J. Zhao, J. Wu, and B. Zhang, "Individualized optimization of handover parameters in 5g (nr) indoor and outdoor co-frequency scenarios," in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021, pp. 59–66.

[157] T. Goyal and S. Kaushal, "Handover optimization scheme for lte-advance

networks based on ahp-topsis and q-learning," *Computer Communications*, vol. 133, pp. 67–76, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366417312549

[158] D. Castro-Hernandez and R. Paranjape, "Optimization of handover parameters for lte/lte-a in-building systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5260–5273, 2018.

[159] M. M. Hasan, S. Kwon, and S. Oh, "Frequent-handover mitigation in ultra-dense heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 1035–1040, 2019.

[160] K. Da Costa Silva, Z. Becvar, and C. R. L. Frances, "Adaptive hysteresis margin based on fuzzy logic for handover in mobile networks with dense small cells," *IEEE Access*, vol. 6, pp. 17 178–17 189, 2018.

[161] T. A. Achhab, F. Abboud, and A. Assalem, "A robust self-optimization algorithm based on idiosyncratic adaptation of handover parameters for mobility management in lte-a heterogeneous networks," *IEEE Access*, vol. 9, pp. 154 237–154 264, 2021.

[162] A. Alhammadi, M. Roslee, M. Y. Alias, I. Shayea, S. Alraih, and K. S. Mohamed, "Auto tuning self-optimization algorithm for mobility management in lte-a and 5g hetnets," *IEEE Access*, vol. 8, pp. 294–304, 2020.

[163] R. Karmakar, G. Kaddoum, and S. Chattopadhyay, "Mobility management in 5g and beyond: A novel smart handover with adaptive time-to-trigger and hysteresis margin," *IEEE Transactions on Mobile Computing*, vol. PP, pp. 1–16, 01 2022.

[164] I. Shayea, M. Ergen, A. Azizan, M. Ismail, and Y. I. Daradkeh, "Individualistic dynamic handover parameter self-optimization algorithm for 5g networks based on automatic weight function," *IEEE Access*, vol. 8, pp. 214 392–214 412, 2020.

[165] "3gpp ts 37.817: "study on enhancement for data collection for nr and endc"."

[166] "Atoll," [Online]. https://www.forsk.com/, accessed: 26 March, 2023.

[167] Y. Ren, J.-C. Chen, and J.-C. Chin, "Impacts of S1 and X2 Interfaces on eMBMS Handover Failure: Solution and Performance Analysis," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6599–6614, 2018.

[168] K. Vasudeva, M. Simsek, D. López-Pérez, and I. Güvenç, "Analysis of handover failures in heterogeneous networks with fading," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6060–6074, 2016.

[169] X. Xu, Z. Sun, X. Dai, T. Svensson, and X. Tao, "Modeling and analyzing the cross-tier handover in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7859–7869, 2017.

[170] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "Syntheticnet: A 3GPP compliant simulator for AI enabled 5G and beyond," *IEEE Access*, vol. 8, pp. 82 938–82 950, 2020.

[171] M. Umar Bin Farooq, M. Manalastas, W. Raza, S. M. A. Zaidi, A. Rizwan, A. Abu-Dayya, and A. Imran, "A data-driven self-optimization solution for inter-frequency mobility parameters in emerging networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 570–583, 2022.