

FEDERAL UNIVERSITY OF SÃO CARLOS– UFSCAR
CENTER OF EXACT SCIENCES AND TECHNOLOGY– CCET
DEPARTMENT OF COMPUTER SCIENCE– DC
POSTGRADUATE PROGRAM IN COMPUTER SCIENCE– PPGCC

Mina Kaviani

**RAHL – A new Residual-based
Adaptive Huber Loss Function for CQI
prediction in 5G networks**

São Carlos
2024

Mina Kaviani

**RAHL – A new Residual-based
Adaptive Huber Loss Function for CQI
prediction in 5G networks**

Thesis presented to the Postgraduate Program in Computer Science
at the Federal University of São Carlos, as part of the requirements
for obtaining the title of Master in Computer Science.

Concentration Area: Distributed Systems and Computer Networks

Supervisor: Fábio Luciano Verdi

Co-supervisor: Jurandy Almeida

São Carlos

2024

Kaviani, Mina

RAHL – A new Residual-based Adaptive Huber Loss Function for CQI prediction in 5G networks / Mina Kaviani -- 2024.
76f.

Dissertação (Mestrado) - Universidade Federal de São Carlos, campus São Carlos, São Carlos
Orientador (a): Fábio Luciano Verdi
Banca Examinadora: Prof. Dr. Fabio Luciano Verdi, Prof. Dr. Hermes Senger, Prof. Dr. Bruno Yuji Lino Kimura
Bibliografia

1. 5G Network . 2. Channel Quality Indicator (CQI). 3. Residual-based Adaptive Huber Loss (RAHL). I. Kaviani, Mina. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática
(SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Arildo Martins - CRB/8 7180



Folha de Aprovação

Defesa de Dissertação de Mestrado da candidata Mina Kaviani, realizada em 19/09/2024.

Comissão Julgadora:

Prof. Dr. Fabio Luciano Verdi (UFSCar)

Prof. Dr. Hermes Senger (UFSCar)

Prof. Dr. Bruno Yuji Lino Kimura (UNIFESP)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

I dedicate this work to my family, who gave me all the support and encouragement to continue and finish this stage of my life.

Acknowledgements

To God who extraordinarily guides my life, allowing me to be better every day.

I would like to extend my deepest gratitude to my parents and my brother, Shahin, Masoud and Kia. Your unwavering love, sacrifices, and support have been the cornerstone of my journey. Despite the physical distance that separates us, your encouragement has always made me feel as if you were right beside me. Thank you for enduring the hardships that came with letting me pursue my dreams far from home. Your strength and resilience have been my guiding light, and your belief in me has given me the wings to soar to new heights. I am forever grateful for the countless sacrifices you have made, the wisdom you have imparted, and the unconditional love you have showered upon me. This achievement is as much yours as it is mine. Thank you for being my inspiration and my pillars of strength.

I would like to extend my heartfelt gratitude to my advisor, Prof. Fábio Verdi. Your guidance, expertise, and unwavering support have been instrumental in the completion of this thesis. From the very beginning, you have been a source of inspiration and motivation, providing invaluable insights and constructive feedback. Your patience, encouragement, and belief in my abilities have helped me overcome countless challenges and grow both academically and personally. This thesis would not have been possible without your guidance, and I am honored to have had you as my advisor.

I would like to extend my gratitude to my co-advisor, Prof. Jurandy Almeida. Your invaluable guidance, support, and expertise have played a crucial role in the successful completion of this thesis. Throughout this journey, your insightful feedback, encouragement, and dedication have been immensely helpful. You have provided me with a fresh perspective and have always been there to offer thoughtful advice and direction. Your willingness to go above and beyond to ensure my success is something I will always be grateful for. I am deeply appreciative of your time and effort, and I am honored to have had you as my co-advisor. This accomplishment would not have been possible without your support.

To my colleagues at LERIS (UFSCar): Leandro, André, Gustavo, Guilherme, Washington, Felipe, Renato, Thiago Henrique, Thiago Caproni, Mateus, Alireza for the camaraderie and the enriching exchange of experiences that have characterized our time together over the past few years. Thank you all for your support and collaboration.

Thank you to everyone who has supported me, whether directly or indirectly, throughout this journey.

“Humility is the first step to wisdom.”
(Saint Thomas Aquinas)

Abstract

The evolution of data consumption patterns, increasingly decentralized from human behavior, has exerted pressure on the transmission resources within the current mobile network infrastructure. In response to this challenge, Fifth Generation Networks (5G) have been standardized and globally deployed, incorporating high-frequency millimeter-wave links to meet the escalating demands for data transfer speeds.

5G signifies the fifth generation of wireless technology, distinguished by its enhanced data speeds, reduced latency, and expanded capacity in comparison to earlier iterations. This advancement holds the potential to transform communication, facilitating the deployment of cutting-edge applications like autonomous vehicles, augmented reality, and seamless Internet of Things (IoT) connectivity. Its extensive implementation is poised to revolutionize various industries and bolster connectivity within our increasingly digitalized environment.

However, the existing management approach for 5G networks is reactive, relying on user devices to calculate network parameters, which are then periodically transmitted to base stations. This reactive methodology introduces delays and network slowdowns, potentially jeopardizing the timely fulfillment of operational requirements.

In the realm of 5G networks, the Channel Quality Indicator (CQI) plays a pivotal role in adjusting modulation and coding schemes based on channel conditions, ensuring optimal data transfer rates and network performance. Recent research focuses on improving CQI estimation in 5G networks using machine learning. In this field, loss functions play a vital role, serving as a guide for training models and optimizing their performance. Two commonly used loss functions are Mean Squared Error (MSE) and Mean Absolute Error (MAE). Roughly speaking, MSE put more weight on outliers, MAE on the majority. Here, we argue that the Huber loss function is more suitable for CQI prediction, since it combines the benefits of both MSE and MAE. To achieve this, the Huber loss transitions smoothly between MSE and MAE, controlled by a user-defined hyperparameter called delta. However, finding the right balance between sensitivity to small errors (MAE)

and robustness to outliers (MSE) by manually choosing the optimal delta is challenging. To address this issue, we propose a novel loss function, named Residual-based Adaptive Residual-based Adaptive Huber Loss (RAHL). In RAHL, a learnable residual is added to the delta, enabling the model to adapt based on the distribution of errors in the data. Our approach effectively balances model robustness against outliers while preserving inlier data precision. The widely recognized Long Short-Term Memory (LSTM) model is employed in conjunction with RAHL, showcasing significantly improved results compared to the aforementioned loss functions. The RAHL has enhanced prediction accuracy for datasets A, B, and C by approximately 11%, 14%, and 0.3%, respectively, compared to the Huber loss function; by 22%, 23%, and 17%, respectively, compared to the MSE loss function; and by 5%, 10%, and 5%, respectively, compared to the MAE loss function. The obtained results affirm the superiority of RAHL, offering a promising avenue for enhanced CQI prediction in 5G networks.

Keywords: 5G Network. Channel Quality Indicator (CQI). Long Short Term Memory (LSTM). Mean Squared Error (MSE). Mean Absolute Error (MAE), Huber Loss, Residual-based Adaptive Huber Loss (RAHL).

List of Figures

Figure 1 – Thesis organization.	29
Figure 2 – Architectural perspective of the internet.	32
Figure 3 – Architectural perspective of the internet.	36
Figure 4 – Structure of a 3G mobile network.	37
Figure 5 – 5G network infrastructure.	39
Figure 6 – Channel Quality Indicator (CQI) in Wireless Communication.	40
Figure 7 – Link Adaptation Model.	40
Figure 8 – How Recurrent Neural Network (RNN) Works.	41
Figure 9 – Most used gated RNN structures. LSTM.	43
Figure 10 – General structure adopted to execute the CQI prediction framework in 5G networks.	52
Figure 11 – General structure adopted to execute the CQI prediction framework in 5G networks.	52
Figure 12 – Considered LSTM scheme.	53
Figure 13 – Exploring how different loss functions shape regression models.	56
Figure 14 – The Huber loss for various values of δ , moving between MSE and MAE.	57
Figure 15 – Splitting a transmission record.	60
Figure 16 – Inference with models trained on normalized and denormalized data.	61
Figure 17 – Sliding windows for time series forecasting.	61
Figure 18 – Network architecture and training hyperparameters for the LSTM model.	62
Figure 19 – Comparison of the ground truth value and the model’s prediction for a sample from the Mobility category of the dataset A.	64
Figure 20 – Comparison of the ground truth value and the model’s prediction for a sample from the Amazon_Prime category of the dataset B.	65
Figure 21 – Comparison of the ground truth value and the model’s prediction for a sample from the dataset C.	66
Figure 22 – Absolute percentage error accumulated over time for different losses.	68

List of Tables

Table 1 – Summary listing the main characteristics of the works cited.	50
Table 2 – MAPE values for the Huber loss with different values for δ	63
Table 3 – MAPE values obtained for different loss functions.	63

List of Acronyms

ARIMA Auto-Regressive Integrated Moving Average technique

AI Artificial Intelligence

BS Base Stations

B5G Beyond 5G Networks

CQI Channel Quality Indicator

CLM Channel Level Metrics

CNN Convolutional neural network

DL Deep Learning

DNN Deep Neural Network

eMBB enable Enhanced Mobile Broadband

eNB evolved Node B

EPC Evolved Packet Core

ELU Exponential Linear Unit

EID Experiment ID

5G Fifth Generation Networks

4G Fourth generation networks

GSM Global System for Mobile Communications

GGSN Gateway GPRS Support Node

GPRS General Packet Radio Service

ISPs Internet Service Providers

IoT Internet of Things

LSTM Long Short-Term Memory

LTE Long-Term Evolution

LA Link adaptation

MSE Mean Squared Error

MAE Mean Absolute Error

mMTC Massive Machine-Type Communications

MAPE Mean Absolute Percentage Error

MIMO Multiple Input Multiple Output

MCS Modulation and Coding Scheme

MMSE Minimum Mean-Square Error

MAC Medium Access Control

MMS Min-Max Scaler

ML Machine Learning

NF Network Functions

NwDAF Network Data Analytics Function

NaN Not a Number

OSI Open Systems Interconnection

QoE Quality of Experience

QoS Quality of service

RAHL Residual-based Adaptive Huber Loss

RSRQ Reference Signal Received Quality

RSRP Reference Signal Received Power

RSSI Received Signal Strength Indicator

RNN Recurrent Neural Networks

RMSE Root-Mean-Squared Error

RI Radio Interface

ReLU Rectified Linear Unit

SNR Signal-to-Noise Ratio

SGSN Serving GPRS Support Node

SDNs Software-Defined Networks

SELU Scaled Exponential Linear Unit

2G Second generation networks

3G Third generation networks

3GPP The 3rd Generation Partnership Project

uRLLC Ultra-Reliable Low-Latency Communications

UE User equipment

VoIP Voice over IP

Contents

1	INTRODUCTION	23
1.1	Problem Statement	25
1.2	Objective	26
1.3	Hypothesis	26
1.4	Justifications	27
1.5	Research Questions and Contributions	27
1.6	Organization of this Thesis	28
2	FUNDAMENTAL CONCEPTS	31
2.1	Computer Networks	31
2.1.1	Network core	34
2.1.2	Unguided networks	35
2.1.3	5G network	37
2.1.4	Channel Quality Indicator (CQI)	39
2.2	Deep Learning (DL)	41
2.2.1	Recurrent Neural Network (RNN)	41
2.2.2	Long Short-Term Memory (LSTM)	42
2.2.3	Most Commonly Utilized Loss Functions	43
2.3	Summary of the Chapter	46
3	RELATED WORK	47
3.1	Summary of the Chapter	50
4	USING RESIDUAL-BASED ADAPTIVE HUBER LOSS FOR PREDICTION CQI	51
4.1	Data set	51
4.2	The deep learning-based CQI prediction model	53

4.3	Huber Loss Function	53
4.4	Residual-based Adaptive Huber Loss (RAHL) - the improved Huber loss for CQI prediction	54
4.5	Predictions of future values	57
5	EXPERIMENTAL EVALUATION	59
5.1	Experimental procedure	59
5.1.1	Data preparation	59
5.2	Training and tuning the parameters of ML	61
5.3	Performance metrics	62
5.4	Results	62
6	CONCLUSION	69
	BIBLIOGRAPHY	73

Chapter 1

Introduction

The deployment of 5G represents a significant leap forward in telecommunications, introducing a trio of services designed to enable Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (uRLLC), and Massive Machine-Type Communications (mMTC). The efficacy of these networks hinges on the proficient management of the 5G Core, where Network Functions (NF) play a pivotal role. To ensure optimal communication, radio signal quality indicators are indispensable in managing 5G links. These indicators, including CQI, Signal-to-Noise Ratio (SNR), Reference Signal Received Quality (RSRQ), Reference Signal Received Power (RSRP), and Received Signal Strength Indicator (RSSI), offer valuable insights into communication link quality. User equipment (UE) collects these indicators and communicates them to the evolved Node B (eNB), which serves as the base station. The eNB's radio network controller adjusts channel modulation based on this information to enhance communication links for UEs.

However, the reactive nature of this process poses challenges, as the collected indicators reflect events in the recent past, and relying solely on reactive operations may not suffice for optimal performance, especially with 5G links characterized by short-range, high-frequency radio signals and mobile UEs. To overcome this limitation and foster more proactive network management, there is a growing need for predictive analytics and machine learning algorithms. These technologies can analyze historical data, identify patterns, and forecast potential issues or changes in the network. Leveraging predictive insights empowers network operators to take proactive measures, optimizing channel modulations, resource allocation, and overall network performance (Yin et al., 2020; Parera et al., 2019; Vankayala; Shenoy, 2020; Sakib et al., 2020; Kimura; Almeida et al., 2021).

In the intricate landscape of 5G networks, effective channel modulation reigns supreme, with the CQI serving as a vital beacon. However, accurately predicting CQI remains a

formidable challenge. The dynamic nature of CQI values, spanning a spectrum from 0 to 15, exhibits abrupt fluctuations dictated by the ever-changing conditions surrounding the UE. Signal quality, oscillating between the pinnacle of 15 and the depths of 4 or even 0, is heavily influenced by physical obstacles like walls, buses, and vehicles.

Noteworthy to say that making a wrong CQI prediction will negatively impact the quality of the 5G channel. The CQI is one of the most important signals used by the base station for modulation and resource allocation. Based on the CQI reported by the UE, the base station allocates bandwidth to maximize the network capacity. This mechanism, if not well configured, will lower the Quality of Experience (QoE) of the end-user. Having a wrong CQI prediction will affect the data rate of each individual application, waste network resources, rapidly deteriorating the end-user experience.

In this context, LSTM has been used as an accurate model for CQI network predictions (Diouf et al., 2022; Bartoli; Marabissi, 2022). Our chosen Machine Learning (ML) model for CQI prediction is the LSTM model, utilizing a singular feature (CQI) and conducting regressions with the one-dimensional temporal series of network information as input data (Bartoli; Marabissi, 2022).

This environmental dynamism poses a significant hurdle in accurately forecasting CQI. Machine learning models, although potent instruments, may encounter challenges in this particular context. Frequently misinterpreting abrupt shifts as outliers, they might struggle to accurately grasp the inherent patterns and trends in the data. Consequently, this limitation can result in diminished performance and compromised accuracy when predicting CQI, making the application of ML in this scenario less optimal. Therefore, the complex dynamics of CQI prediction within the ever-changing 5G ecosystem emphasize the necessity for alternative methodologies. The decision regarding whether to employ MSE or MAE as the error metric for CQI signal quality evaluation is rendered complex by the inherent limitations of both metrics and the necessity of factoring in the specific conditions affecting CQI signal accuracy.

To address this, we explore the Huber loss function, known for its robust, piece-wise structure that mitigates the influence of outliers compared to MSE (Khan; Wang; Yang, 2016), (Raca et al., 2020). Huber loss function (Gokcesu; Gokcesu, 2021) seamlessly blends the quadratic (MSE) and absolute value (MAE) losses, offering a user-controlled trade-off via a hyperparameter called delta (δ). However, manually setting this hyperparameter to balance sensitivity to small errors (MAE) and robustness to outliers (MSE) can be challenging. Motivated by this, instead of manually setting this hyperparameter, which is hard, we propose the RAHL, which transforms the hyperparameter delta into a trainable parameter. By transforming delta into a trainable parameter, RAHL empowers models to learn optimal outlier robustness during training, achieving a sweet spot between outlier resistance and inlier accuracy (Dong; Yang, 2020; Gokcesu; Gokcesu, 2021).

The Huber Loss is a robust combination of the MSE and MAE loss functions. It

introduces a hyperparameter δ , that determines the transition point between the two functions. When the absolute difference between the predicted and true values is smaller than δ , Huber Loss behaves like MSE. Otherwise, it transitions to a linear MAE-based function. This dual nature helps in balancing the sensitivity to outliers and the stability of gradient-based optimization (Jadon; Patil; Jadon, 2024).

Through a comprehensive investigation and together with the LSTM model for CQI prediction, we systematically evaluated the impact of RAHL in model training, comparing its performance to alternative loss functions like the standard Huber, MSE and MAE. Employing Mean Absolute Percentage Error (MAPE) as the evaluation metric, our results revealed that the RAHL consistently produced lower MAPE values compared to other loss functions, indicating improved model accuracy. This research contributes to a deeper understanding and broader application of machine learning models in forecasting signal quality indicators, ultimately leading to enhanced 5G network performance.

1.1 Problem Statement

In machine learning models, the choice of an appropriate loss function is paramount for precise estimation and evaluation. The consistency in selecting a loss function holds significance in two crucial aspects. First, within a model, the loss function used for parameter estimation must align with the one employed for model evaluation to guarantee efficient optimization and accurate assessment. Second, when comparing different models, employing consistent loss functions becomes imperative for meaningful comparisons. Discrepancies in this regard can result in erroneous performance evaluations and can add complexity to the decision-making process when determining the superior model (Christoffersen; Jacobs, 2004). A loss function plays a pivotal role in assessing parameter performance during the model-fitting process to the data. It quantifies the disparity between the model's predictions and the desired output (Khan; Wang; Yang, 2016). In the domain of statistical analysis, the MSE is highly susceptible to outliers. A single outlier with a substantial error can significantly inflate the MSE, making it an inadequate choice for datasets containing such exceptional data points (Raca et al., 2020). Furthermore, it is important to note that the MSE is sensitive to model assumptions, especially the assumption of normally distributed errors with constant variance, as violations of these assumptions can introduce bias into parameter estimates (Fearn, 2012). In the realm of regression problems, the absolute loss, $L_1(y, f_\theta(x)) = |y - f_\theta(x)|$, and the ubiquitous quadratic (squared) loss, $L_2(y, f_\theta(x)) = (y - f_\theta(x))^2$, emerge as consequential alternatives, requiring a strategic choice rooted in their distinctive characteristics. The quadratic loss, recognized for its robust convexity, facilitates rapid learning rates, while the absolute loss is esteemed for its inherent resilience. This dichotomy emphasizes the need of amalgamating the strengths of both loss functions, leading to models that not only exhibit robustness

against outliers but also achieve swift convergence with optimal goodness of fit.

The Huber loss emerges as a prevalent solution, seamlessly combining quadratic and absolute losses to formulate a resilient loss function that converges quickly (Gokcesu et al., 2018). The Huber loss is widely adopted in regression tasks, especially when dealing with outliers or noise in the data. It achieves a balanced compromise between MSE (quadratic loss) and MAE (absolute loss), offering improved resilience against extreme data points compared to relying solely on MSE or MAE. The pivotal transition point within the Huber loss dictates its shift from quadratic to absolute loss behavior, making it a crucial hyperparameter that significantly influences the performance of a regression model. Nevertheless, the challenge lies in selecting the optimal transition parameter, necessitating frequent hyperparameter searches to identify the most suitable value (Meyer, 2021).

Tuning the hyperparameter δ by hand poses substantial challenges for adopting the Huber loss to train regression models, which include subjectivity, data dependence, computational burden, potential overfitting, and the difficulty of balancing accuracy and robustness. This requires a cautious approach and exploration of more sophisticated hyperparameter tuning methods for robust regression models. To mitigate these shortcomings, we propose the RAHL, a transformative approach that empowers the model to automatically determine the optimal penalty scheme.

1.2 Objective

The objective of this study is to address the challenges associated with manually tuning the transition parameter δ in the Huber loss for training regression models. We aim to propose and evaluate the efficacy of a novel approach, the RAHL, which enables the model to autonomously determine the optimal penalty scheme. This research seeks to enhance the robustness and efficiency of regression models by leveraging an adaptive loss function that dynamically adjusts to the characteristics of the data. Through empirical evaluation and comparison with existing methods, we aim to demonstrate the effectiveness of RAHL in achieving a balance between accuracy and robustness while mitigating the subjective and computational burdens associated with manual hyperparameter tuning. This study contributes to the advancement of machine learning techniques by offering a more automated and data-driven approach to loss function selection, thereby facilitating improved model performance and generalization across diverse datasets.

1.3 Hypothesis

The RAHL method has the potential to significantly improve regression models by offering both robustness against outliers and competitive accuracy on various datasets.

Unlike traditional approaches that require manual tuning of the Huber loss function, RAHL dynamically adjusts this hyperparameter based on residual analysis. This not only eliminates the need for subjective and computationally expensive tuning procedures but also enhances the generalizability of the model across different datasets and tasks. As a result, RAHL is expected to lead to more reliable and efficient regression models compared to conventional methods.

1.4 Justifications

There are various approaches that address objectives similar to those of this work, specifically related to CQI regression using deep learning methods in Beyond 5G Networks (B5G) networks—more information can be found in Chapter 3 of this document. However, the following open points can be highlighted from other works in the literature:

1. Absence of real datasets, such that their models are trained and evaluated only on synthetic data and simulated environments. Model adjustments or even the definition of environments that use only other types of mobile networks, such as Fourth generation networks (4G) networks.
2. Previous works often use common loss functions, such as MSE or MAE, for CQI regression but may lack models adjusted for real data and diverse environments, like 5G networks.

This work focuses on predicting the CQI, which is crucial for optimizing 5G communication systems. Beyond developing methods for accurate CQI prediction, this study highlights the importance of selecting appropriate loss functions that are tailored to datasets characteristic of 5G environments. We explored the effects of various loss functions, including Huber loss, MSE, MAE, Quantile, and LogCoshLoss, and discovered that fine-tuning the delta hyperparameter in the Huber loss function can significantly improve prediction accuracy and reliability. By carefully choosing loss functions that best suit these specific datasets, this research aims to enhance the overall performance of prediction models, thereby providing more accurate and valuable results. Consequently, this study addresses both the fundamental challenge of CQI prediction and the technical aspect of improving prediction accuracy through effective loss function selection.

1.5 Research Questions and Contributions

In this section, we address the two research questions that guide this thesis.

Question 1: What factors led to your selection of CQI as the primary indicator for prediction in the context of 5G networks?

The CQI serves as a critical parameter in Long-Term Evolution (LTE) networks. It critically manages channel variations by accurately assessing channel conditions and noise variance estimations. However, the implementation of Adaptive Modulation and Coding in 5G and Beyond 5G Networks introduces complexities. The inherent delay in the feedback loop can render the CQI outdated, leading to significant degradation in user communication (Balieiro; Dias; Guarda, 2021). Nevertheless, within communication system design, the CQI plays a pivotal role. It encapsulates the channel's current state and empowers Base Stations (BS) to dynamically tailor the quality of service to match prevailing conditions, thus ensuring seamless and efficient communication (Vankayala; Shenoy, 2020).

Question 2: What is the reason behind optimizing the Huber loss function and developing a novel loss function?

The Huber loss is a widely adopted loss function in regression tasks, especially in situations where the data may include outliers or noise. It strikes a harmonious balance between the MSE (quadratic loss) and the MAE (absolute value loss), rendering it more resilient to extreme data points compared to relying solely on mean squared error. The transition point within the Huber loss determines when it transitions from a quadratic to an absolute value behavior, making it a crucial hyperparameter that significantly impacts the performance of a regression model. Nevertheless, selecting the optimal transition parameter, which governs the shift between quadratic and absolute value behavior, has historically presented a challenge, necessitating frequent hyperparameter searches to find the best value (Meyer, 2021). Choosing the right value for δ is crucial, as it determines the transition point where the loss function switches from prioritizing precision to emphasizing robustness. A smaller δ favors MAE behavior, boosting accuracy but decreasing resilience to outliers. Conversely, a larger δ pushes the loss function toward MSE characteristics, enhancing robustness but potentially at the cost of accuracy.

Tuning the hyperparameter δ by hand poses substantial challenges for adopting the Huber loss to train regression models, which include subjectivity, data dependence, computational burden, potential overfitting, and the difficulty of balancing accuracy and robustness. This requires a cautious approach and exploration of more sophisticated hyperparameter tuning methods for robust regression models. To mitigate these shortcomings, we propose the RAHL, a transformative approach that empowers the model to automatically determine the optimal penalty scheme.

1.6 Organization of this Thesis

The organization of the chapters, their correlation and the placement of each component are detailed in Fig. 1. After the introduction, the fundamental concepts necessary for an understanding of the topics discussed in this thesis are presented in Chapter 2. In

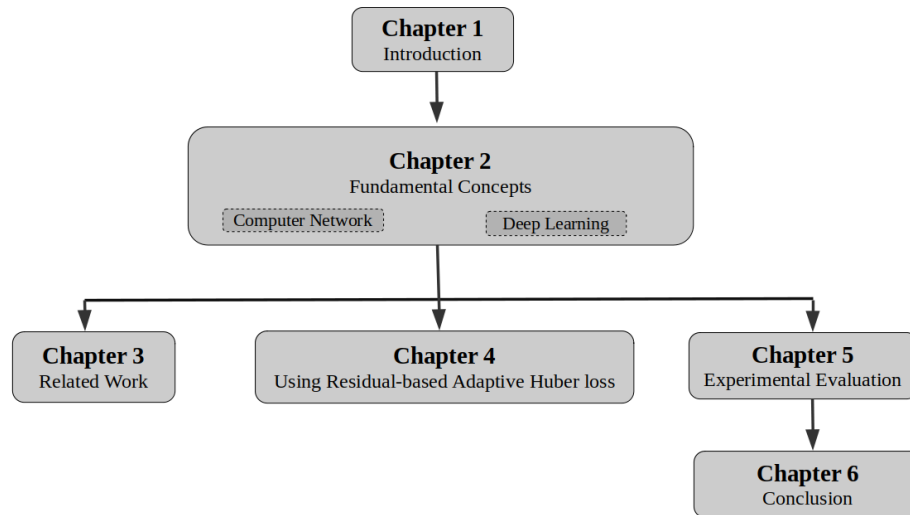


Figure 1 – Thesis organization.

Chapter 3, the related works are briefly described, providing an exposition of the primary distinctions achieved through the advancements introduced within this thesis. Chapter 4 offers a comprehensive guide on creating the RAHL and provides detailed instructions on how to implement it effectively. Chapter 5 presents the experimental evaluation and results of applying various loss functions, along with a comparative analysis of their effectiveness. Chapter 6 presents the final considerations, including lessons learned and future directions to be explored.

Chapter 2

Fundamental Concepts

In this section, our aim is to elucidate the fundamental concept that forms the core of our proposal, providing a comprehensive grasp of its significance and its relevance to our project, highlighting how the utilization of the new loss function (RAHL) indisputably enhances the accuracy of CQI prediction. our solution can be interpreted as a fusion of machine learning model and finding best Loss function for CQI prediction.

2.1 Computer Networks

In this section, we introduce concepts and characteristics of computer networks, particularly those related to the global internet. The majority of the discussion in this chapter has been drawn from and adapted from Kurose and Ross (2016), except where explicitly referenced by other authors. Kurose is a renowned professor in the field of networking at the University of Massachusetts, with his book *Computer Networking: A Top-Down Approach* serving as the foundation for numerous networking courses worldwide.

Kurose and Ross (2016), describe the internet as a complex network of interconnected systems spanning the globe. Beyond its initial purpose of linking computers, it now seamlessly integrates a dazzling array of devices, from smart TVs and phones to internet-connected washing machines. This ever-expanding ecosystem relies on diverse communication links, acting as the highways for digital information. These highways come in two flavors: guided media, like the familiar wires and cables, and unguided media, using invisible waves like Wi-Fi or even light itself. Each has its own speed limit, but external factors like interference and congestion can throw a wrench in the data flow, slowing things down. In essence, the internet is a marvel of connection, constantly evolving to

accommodate our ever-growing appetite for information and digital interaction, all while battling the challenges of speed and reliability.

In computer networks, communication between devices (called end systems) occurs through messages broken down into smaller units called packets. These packets, like individual puzzle pieces, carry not only the message itself but also essential information such as destination addresses, size, and protocols used. Think of this information as labels and instructions written on each piece to ensure it reaches the right place and assembly happens smoothly. Once all packets arrive at the destination, their 'labels' are removed, and the original message is reconstructed. Additionally, since devices might not be close enough for a direct connection, these messages travel through a vast network of interconnected links, similar to how roads help deliver packages across great distances. This intricate system of packets and networks ensures efficient and reliable information exchange between devices, even across the globe.

The internet can be viewed from two perspectives: (1) a network architectural perspective, where Internet Service Providers (ISPs) globally connect various end systems through a network of Routers (Packet Switches) and Link Switches connected by links (Figure 2 - Source: Adapted from (Kurose; Ross, 2016)); or (2) from a services perspective, which sees the internet merely as a (large) infrastructure that provides services (i.e., communication) to applications distributed among different end systems.

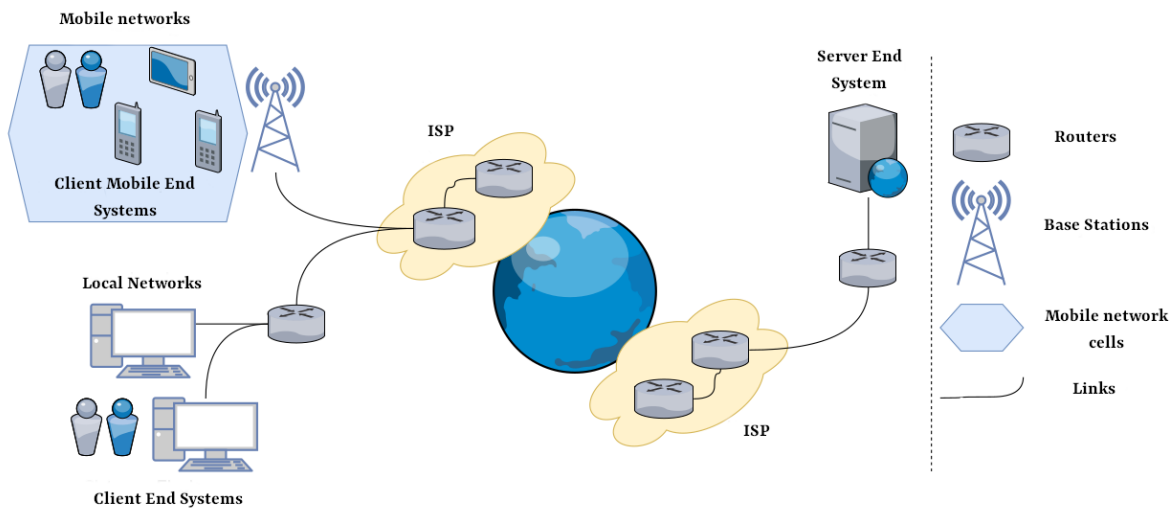


Figure 2 – Architectural perspective of the internet.

The internet, a marvel of global connection, relies on standardized protocols to enable communication between diverse devices and systems. These protocols act like pre-arranged languages, dictating how information is formatted and exchanged. Imagine it like a layered cake, each layer representing a specific protocol with its own function. Messages travel through these layers, getting encapsulated and processed at each stage. This layered architecture ensures modularity, allowing changes to one layer without impacting the others, as long as the communication interfaces remain consistent. It's this standard-

ization and layered approach that allows devices from different corners of the world to seamlessly connect and share information, forming the interconnected web we know and use today.

The internet can be divided into 5 main layers, namely the physical layer, the data link layer, the network layer, the transport layer, and the application layer. This set of layers is commonly referred to as the internet protocol stack. In their book, Kurose and Ross (2016) study each of the layers in a top-down manner, which are presented as follows:

- ❑ **Application Layer:** Consists of a protocol distributed across different end systems, which allows the application of one end system to exchange messages with the application on another end system. Examples of application layer protocols include HTTP, SMTP, FTP, among others.
- ❑ **Transport Layer:** The transport layer carries messages from the application layer and encapsulates them in the form of segments. There are two main protocols for the transport layer, TCP and UDP. TCP provides connection-oriented services to applications, such as guaranteed message delivery, flow control, message fragmentation, and congestion control. UDP, on the other hand, does not provide connection-oriented services, referring to a more simplified service that guarantees security instead of flow control and congestion control.
- ❑ **Network Layer:** This layer's primary protocol is the IP protocol and its routing protocols. IP is a protocol adopted by the entire internet, which allows the identification and addressing of end systems. IP first encapsulates segments into datagrams, which contain various fields in their headers, such as addressing and other control information that may indicate how intermediate nodes (routers) and end systems should act on these received datagrams.
- ❑ **Data Link Layer:** Although the network layer encapsulates datagrams and defines the path that the packet should follow, it is not the layer that actually sends the datagram to the next node on the path. This is done by the data link layer. The received datagram is encapsulated in frames and forwarded by the link connecting to the next node. The protocols of this layer are strongly associated with link communication technology (for example, Ethernet protocol and WiFi protocol), with the main function of preparing datagrams to be launched onto the link in question. Additionally, as datagrams can transit through various nodes, different link protocols can be used along the transmission path of a message.
- ❑ **Physical Layer:** The physical layer is the one that effectively transmits the bits of the frames through physical media in the real world (guided and unguided media).

There are also other protocol stacks such as the Open Systems Interconnection (OSI) model defined by ISO, which have additional session and presentation layers beyond the

standard 5-layer model (totaling seven layers). Strictly speaking, the addition of these layers depends on the application's needs and may therefore be included in the application layer.

In general, not all equipment that makes up the network implements all layers of the protocol stack. End systems where distributed applications are executed must necessarily execute all layers, but this is not necessary for other network equipment, such as link switches that only contain the physical layer and the data link layer, and routers reaching up to the network layer.

2.1.1 Network core

Viewed through the lens of service delivery, the internet acts as a complex network infrastructure that facilitates message exchange between disparate end systems. These messages are broken down into data packets and transmitted across interconnected nodes, including packet-switching devices and link switches. The network encompasses edge networks responsible for connecting individual devices, and core networks that enable global connectivity between these edge networks, ultimately establishing pathways for data to travel from its source to its intended destination. Figure 2 visually depicts both components, with local networks, mobile networks on the left, and server connections on the right representing edge networks, while the regional and international networks within the ISP illustrate the core network infrastructure.

Ideally, packet transmission between applications would occur instantaneously, but this is impossible through classical means due to various factors such as physical limitations, external interferences, protocols, and network congestion. For example, network nodes typically adopt the store-and-forward transmission system, waiting for the receipt of all bits of a packet before analyzing its header and forwarding it to the next node in the network. Additionally, links transmit only one piece of information (bits) from the packets at a time, causing each node to maintain a queue structure (first in, first out) to store received packets until the outgoing link is free for transmission. Due to these factors, packet transfer can experience delays that may diminish application performance. Delays can be categorized as nodal delays, which occur due to the operation mode of the nodes, such as delays in Processing, queuing, transmission, and propagation, as well as end-to-end delays (latency), encompass the accumulation of all delays that may occur along the entire path of the packet in the network. There are also other delays that applications may add to the process, such as packaging delay in Voice over IP (VoIP) applications and delay induced by multimedia transmission systems.

2.1.2 Unguided networks

Wireless hosts are essential components of 5G networks, and understanding their definitions is crucial for a deeper grasp of this technology.

- **Wireless hosts:** Unwired end systems, such as cell phones and laptops.
- **Wireless links:** Unguided physical media that transmit bits via electromagnetic waves.
- **Base station:** Specific devices responsible for relaying and coordinating packets sent to and from wireless hosts associated with them (i.e., connected and, consequently, within their range). Examples of base stations include cellular towers for mobile networks (referred to as node B - NB for Third generation networks (3G)/4G networks and evolved Node B - eNB for 5G networks) and WiFi access points typically found in local networks following the IEEE 802.11 standard.

Mobile networks were conceived with the intention of providing internet connectivity to mobile hosts, and since cellular telephony was already present in much of the world, a possible strategy would be to extend it to also handle connection and packet transfer with the internet network. The first standard developed was the Global System for Mobile Communications (GSM), considered the precursor of mobile networks that identified and solved the problem of incompatible communications between different parts of Europe. Subsequently, various cellular communication technologies began to be defined and deployed, always seeking to increase coverage area and provide better connectivity and increasingly higher data transfer rates. These mobile networks are typically divided into generations, which are: 1st generation, focused solely on voice traffic; 2nd generation, also focused on voice, but later extended to data transmission with its 2.5 version; 3rd generation, the first generation that originally supported both forms of voice and data communication; 4th generation, eliminated the distinction between voice and data networks of the 3G core; and currently, 5th generation networks that take into account new patterns of consumption not centralized on human behavior. All generations operate in licensed frequency bands, requiring telecommunications operators to pay large sums of money to governments.

The term cellular comes from the fact that geographic coverage areas are divided into several cells, usually illustrated as hexagons. In general, base stations are inserted at the center of each cell, or at the intersection of three cells, which act as transmission bridges between mobile hosts within the cell's coverage area and the external world internet. In Second generation networks (2G) networks, a set of base stations is coordinated by a single base station controller, which is responsible for allocating channels to subscribers (mobile hosts) and indicating to which base station the hosts are connected. Above the base station controllers, there are also mobile switching centers, which perform user

accounting and authorization. The general architecture of mobile networks can be seen in Figure 3 - Source: Adapted from (Kurose; Ross, 2016).

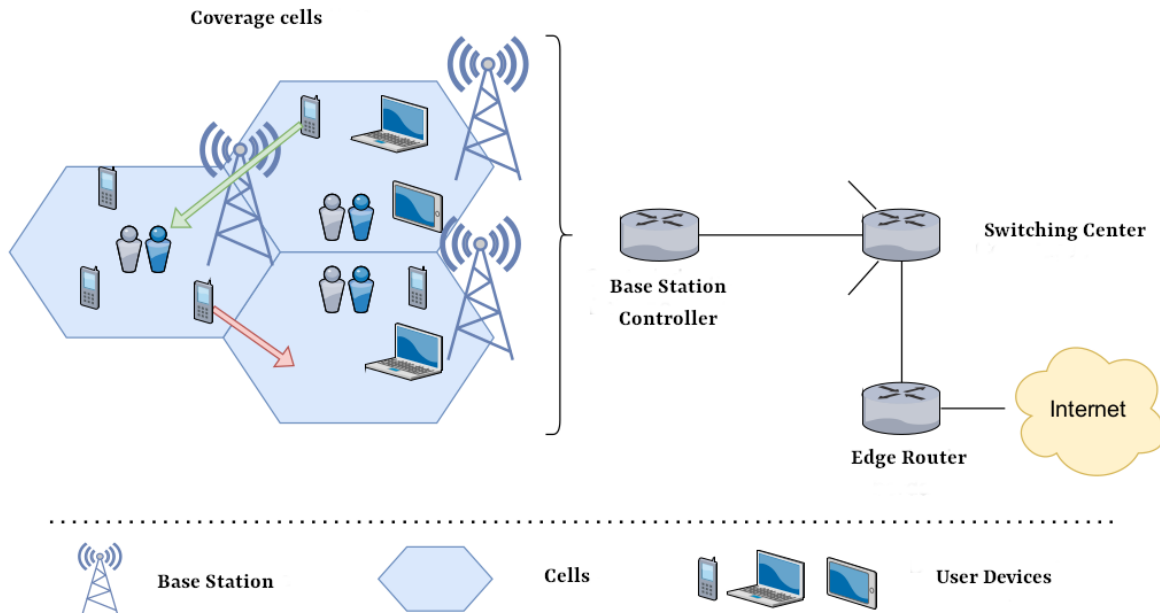


Figure 3 – Architectural perspective of the internet.

The 3G networks (3rd generation) were the first mobile network infrastructure designed from the ground up to handle both voice calls and data transmissions. To achieve this, two main changes were made compared to the previous technology: the edge networks (coverage cells associated with base stations) were modified to handle both GSM voice call service and internet data transmission, and the core of mobile networks was modified with a new component specifically dealing with data transfer, while keeping the previous GSM structure intact to operate solely with voice calls. To accomplish this, two new types of nodes were developed: a Serving GPRS Support Node (SGSN) and a Gateway GPRS Support Node (GGSN). General Packet Radio Service (GPRS) stands for Generalized Packet Radio Service, referring to the old data service of 2G networks. In this new infrastructure of 3G networks (Figure 4 - Source: Adapted from (Kurose; Ross, 2016), SGSNs are responsible for delivering datagrams to mobile hosts associated with the wireless access network, while GGSNs act as edge routers between the SGSN and the external internet network, thereby hiding the mobility of hosts from the internet network.

4G network development, also known as LTE, ushered in a significant leap compared to its predecessor, 3G. Two key innovations were the introduction of the Evolved Packet Core (EPC) and the LTE radio access network. The EPC revolutionized network architecture by unifying voice and data services into a single, IP-based network, streamlining communication and offering increased flexibility. Meanwhile, the LTE radio access network employed a dynamic combination of time and frequency division multiplexing, coupled with Multiple Input Multiple Output (MIMO) antennas, to significantly boost

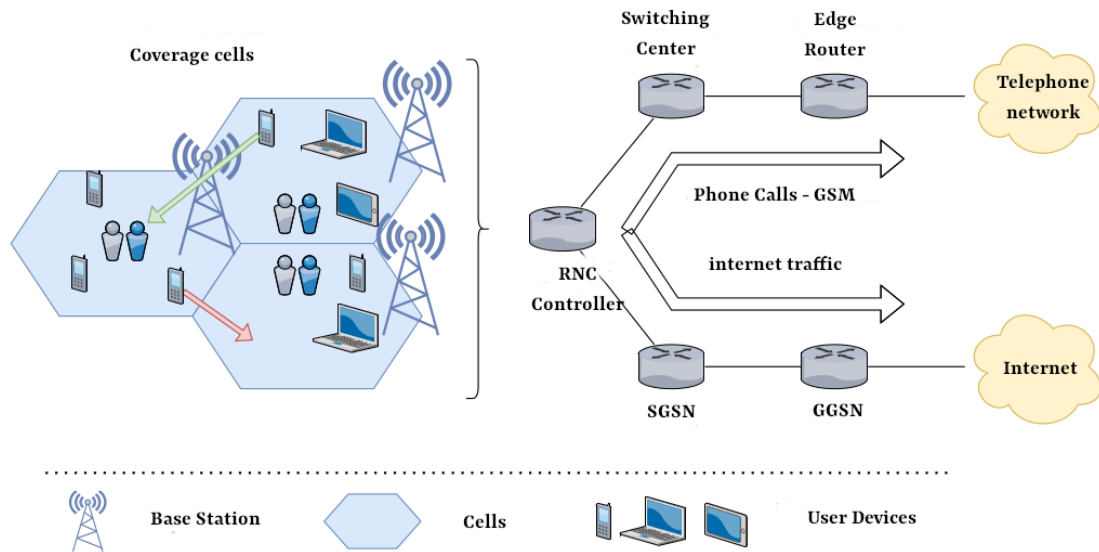


Figure 4 – Structure of a 3G mobile network.

data transfer speeds and network capacity. While some experts raised concerns about VoIP suitability within the IP protocol, the EPC's resource control capabilities aimed to ensure consistently high-quality voice calls. Moreover, the clear separation of control and data planes within the EPC enhanced network efficiency and security. These advancements have solidified 4G's position as the dominant mobile network technology worldwide, forming the foundation for the even more powerful 5G networks of today.

2.1.3 5G network

Mobile networks are currently dominated by 4G technology, originally designed for human users. However, the landscape is rapidly shifting with the emergence of diverse Internet-connected devices like smartwatches, drones, industrial sensors, and medical equipment. These varied IoT applications require different functionalities and network characteristics. Experts like Cisco predict a massive surge in data traffic, potentially tripling between 2017 and 2022, reaching 122 exabytes per month by 2022. This necessitates accommodating billions of simultaneously connected devices, compared to 2017's 11.51 exabytes and a smaller user base. This highlights the need for future mobile network architectures to evolve beyond their initial human-centric design and adapt to the diverse demands of an ever-expanding and interconnected world (Pham et al., 2020; Forecast et al., 2019). Transmission rates at these proportions are approaching the thresholds of the existing network infrastructure, leading to congestion and subsequently, a decline in Quality of service (QoS), which encompasses delivery speed (bandwidth), and QoE, encompassing network latency and delays, as observed by users (Sevgican et al., 2020). Consequently, the development of 5G mobile networks aims to address the evolving demands of internet usage by meeting various criteria, including offering high transfer rates

with minimal latency. This enhancement is poised to significantly elevate both QoS and QoE for users. The 5G networks stand out as heterogeneous systems, delineated into three distinct user categories each exhibiting unique consumption behaviors: eMBB service, uRLLC service, and mMTC. The eMBB service serves to cater to the needs of high data transmission rates, facilitating the seamless operation of high-resolution multimedia applications and cutting-edge virtual/augmented reality devices. On the other hand, uRLLC is primarily tailored to applications necessitating minimal transmission latency, such as critical systems like autonomous vehicles and life-saving telemedicine services. Lastly, mMTC addresses the domain of machine-to-machine communication, characteristically involving a plethora of IoT devices engaged in constant, small-scale data exchanges throughout the day. This starkly contrasts with human behavioral patterns, which exhibit cyclical and seasonal consumption tendencies, often peaking at specific times of the day (Sevgican et al., 2020; Pham et al., 2020). Effectively managing the transfer of various data types within tight time constraints places a significant strain on communication and computational resources (Pham et al., 2020). Consequently, 5G networks have evolved into Software-Defined Networks (SDNs). This transformation has endowed 5G networks with a centrally controlled cloud services infrastructure, thereby virtualizing tasks through NF and substantially alleviating resource demands on eNBs. Nonetheless, 5G networks encounter persistent challenges, including the integration of data and control planes at the entrance of cloud systems (Fu et al., 2018), and performance degradation stemming from the reactive nature (traditional approach) of their edge equipment (eNBs). An example of 5G network infrastructure is indicated in Figure 5. The eNBs (base stations) of current 5G network infrastructures feature MIMO technologies, establishing connections that transmit data through multiple antennas. However, since millimeter-wave electromagnetic waves struggle to penetrate dense materials and have limited coverage areas (with a range of approximately 30 meters), eNBs connect to UE via Small Cells. Additionally, 5G networks separate communication into two planes: a control plane, where network information and control signals are exchanged between the core network and infrastructure equipment, and a user plane, where user packets and messages are transported to and from the internet. It is also noted the dedicated cloud infrastructure that 5G networks possess, which receive requests from eNBs and other equipment and process them through the virtualization of NFs (Gupta; Jha, 2015).

To overcome the limitations of reactive approaches, The 3rd Generation Partnership Project (3GPP) standardization body has developed the Network Data Analytics Function (NwDAF). This key element of B5G networks leverages pre-implemented Artificial Intelligence (AI) algorithms, accessible through requests or triggered by other system components. By analyzing various network parameters, NwDAF offers crucial functionalities, including: identifying abnormal behaviors, predicting expected network performance, monitoring slice load levels, and analyzing network function load and congestion.

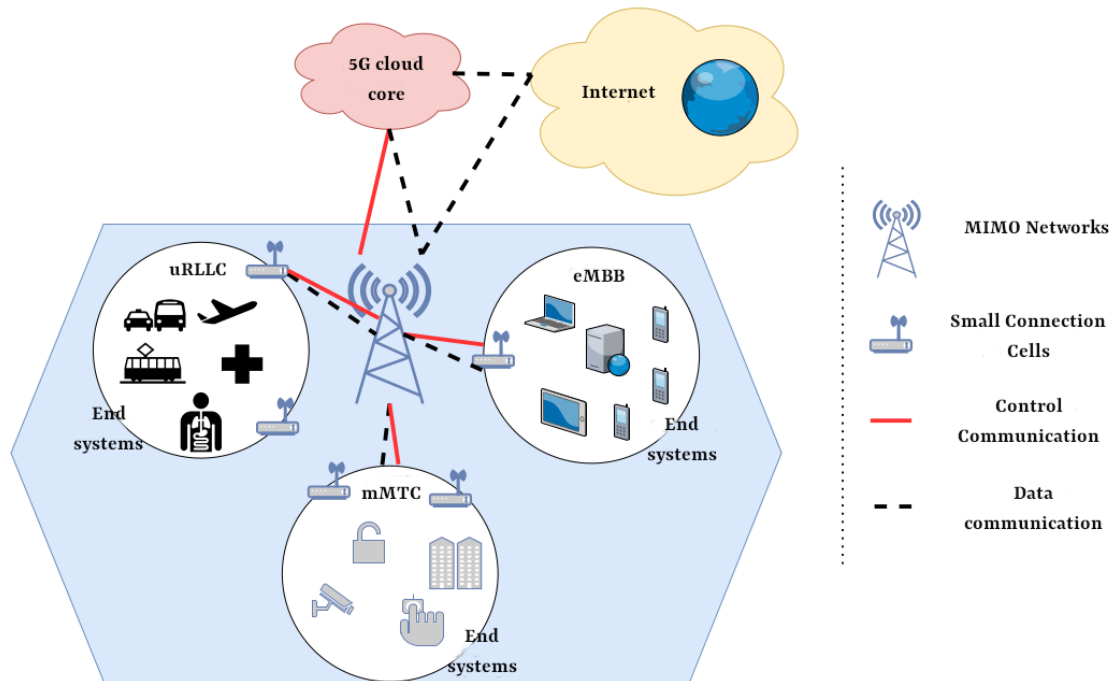


Figure 5 – 5G network infrastructure.

Additionally, it provides insights into mobility patterns and overall quality of service, enabling proactive network management and optimization (Sevgican et al., 2020).

The CQI plays a crucial role in 5G networks, the CQI is a parameter that reflects the quality of the wireless channel between a user device (such as a smartphone) and the base station (cell tower) and It provides information about the reliability and capacity of the channel.

2.1.4 Channel Quality Indicator (CQI)

In the design of the communication system, the CQI depicted plays a pivotal role as a key parameter responsible for encoding the channel's current status. This information enables a BS to dynamically adjust the quality of service in accordance with the prevailing channel conditions, thereby ensuring seamless and efficient communication Figure 6, (Vankayala; Shenoy, 2020).

The CQI plays a crucial role in LTE networks, aiding in the management of channel fluctuations. The precision of CQI computation relies heavily on accurate estimations of both the channel and noise variance (Mansour; Nada; Mehana, 2015).

By utilizing the CQI, the LTE transceiver can effectively adjust to changes in the channel, optimizing system performance by selecting the suitable code rate and modulation order indicated by the CQI index. This index, calculated by the UE, ensures that the block error rate remains below 10% (Rinne et al., 2009).

The CQI serves as an index mirroring the channel's condition. UEs convey this index to their respective cells within the LTE framework, covering a spectrum from 0 to 15.

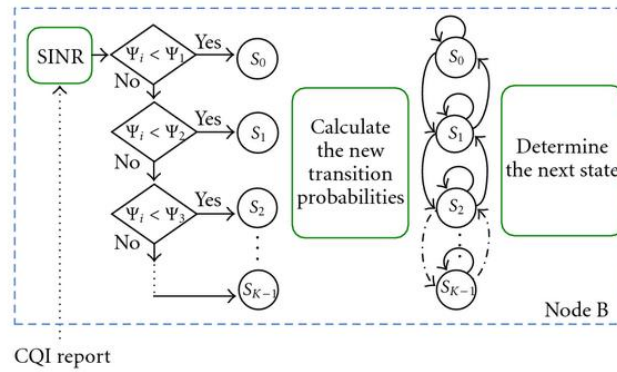


Figure 6 – Channel Quality Indicator (CQI) in Wireless Communication.

Traditionally, CQI estimation relies on historical channel data to forecast its forthcoming quality (Parera et al., 2019).

Link adaptation (LA) is a common technique utilized to manage fluctuations in the channel, as depicted in Figure 7 - Source: Adapted from (Mansour; Nada; Mehana, 2015). LA adjusts the transmission rate in response to changes in the channel by selecting an appropriate Modulation and Coding Scheme (MCS) based on feedback received from the UE. The CQI serves as a vital metric in LTE systems, enabling effective LA. The UE conducts various measurements, including channel and noise variance estimations, to calculate the CQI, which is then relayed back to the base station. Reliable channel and noise variance estimates are essential for the UE to provide an accurate CQI estimate. The accuracy of noise variance estimation, or SNR estimation, impacts several system functionalities, such as adaptive modulation, turbo coding, channel estimation, and equalization techniques like the Minimum Mean-Square Error (MMSE) (Mansour; Nada; Mehana, 2015).

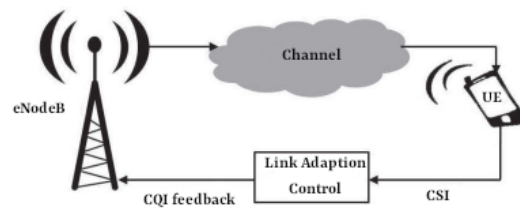


Figure 7 – Link Adaptation Model.

In the domain of CQI prediction, a promising avenue to explore is the application of machine learning methodologies, particularly Recurrent Neural Networks (RNN). These advanced models harness the power of historical data and network conditions to generate accurate CQI forecasts, thereby optimizing the performance and reliability of our 5G networks.

The majority of groundbreaking achievements in the realm of RNNs have been credited to LSTM networks, positioning them as the central focus within the domain of deep learning. Thanks to their exceptional learning capabilities, LSTMs have demonstrated

remarkable performance and have been widely adopted across a diverse array of applications, including, but not limited to, speech recognition (He; Droppo, 2016), acoustic modeling (Qu et al., 2017), trajectory prediction (Alth e; Fortelle, 2017), sentence embedding (Palangi et al., 2015), and correlation analysis (Mallinar; Rosset, 2018).

2.2 Deep Learning (DL)

Deep Learning (DL) represents a specialized area within machine learning focused on the efficient processing and analysis of unstructured data. By leveraging sophisticated neural network architectures, deep learning models progressively learn and extract features from data through multiple layers of abstraction. The superior performance of deep learning techniques over conventional machine learning approaches can be attributed to the growth of available data and significant improvements in computing hardware (Mathew; Amudha; Sivakumari, 2021).

2.2.1 Recurrent Neural Network (RNN)

A RNN is a type of artificial neural network designed for processing sequences of data. Unlike feedforward neural networks, which have a strictly layered structure, RNNs have connections that loop back on themselves, as illustrated in Figure 8 - Source: Adapted from (Li et al., 2019), This loop allows them to maintain a kind of memory of previous inputs. Each step in the sequence is processed along with information from previous steps, making them well-suited for tasks like time series prediction, natural language processing, speech recognition, and more (Medsker; Jain, 2001).

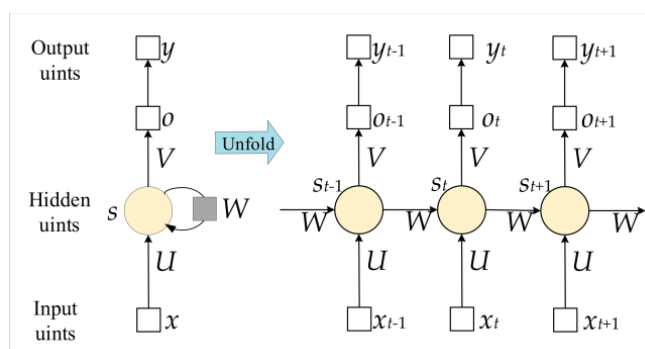


Figure 8 – How Recurrent Neural Network (RNN) Works.

The dynamical system is defined by:

The hidden state at time step t , denoted as h_t , is computed using a state transition function. A RNN is a type of artificial neural network designed for sequential data processing. Unlike traditional feedforward neural networks, where information flows in one direction (from input to output), RNNs have connections that loop back on themselves. This looping mechanism allows RNNs to maintain a form of memory or context, making them

well-suited for tasks involving sequences, time series data, and natural language processing. The function f_h , which depends on the input x_t at time t and the previous hidden state h_{t-1} . This dynamic evolution is expressed as:

$$h_t = f_h(x_t, h_{t-1}) \quad (1)$$

The output at time step t , represented as y_t , is generated through an output function f_o , based on the current hidden state h_t :

$$y_t = f_o(h_t) \quad (2)$$

Both of these functions are parametrized by specific sets of parameters θ_h and θ_o , which we intend to learn from the data. To achieve this, we utilize a training dataset D consisting of N sequences, each containing pairs of input-output sequences $(x_1^{(n)}, y_1^{(n)}), \dots, (x_{T_n}^{(n)}, y_{T_n}^{(n)})$.

Our goal is to optimize the RNN's parameters by minimizing the cost function $J(\theta)$, which measures the dissimilarity between the predicted outputs and the ground truth outputs in the training data. The cost function is defined as:

$$J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} d(y_t^{(n)}, f_o(h_t^{(n)}))$$

where $h_t^{(n)}$ is the hidden state computed using the state transition function, and $h_0^{(n)}$ is initialized to 0. The function $d(a, b)$ represents a predefined divergence measure, which can be chosen based on the specific problem, such as Euclidean distance or cross-entropy (Pascanu et al., 2013).

2.2.2 Long Short-Term Memory (LSTM)

RNNs have found widespread use in various research domains dealing with sequential data, including text, audio, and video analysis. However, traditional RNNs have the limitations of capturing long-term dependencies in sequential data, which is essential for predicting CQI values accurately (Bartoli; Marabissi, 2022). To address this issue and enable the modeling of long-term dependencies, LSTM networks were introduced (see Figure 9 - Source: Adapted from (Bartoli; Marabissi, 2022)). Since their inception, LSTM networks have garnered significant attention and have been instrumental in achieving remarkable results in the field of deep learning (Yu et al., 2019).

The specifications of the LSTM model is as follows:

LSTM controls the information flow through three gates: forget (f) (Equation 3), input (i) (Equation 5), and output (o) (Equation 4), along with two state variables, h and s . Each gate is influenced by the concatenation of the current input with the previous output. It consists of an affine transformation, a sigmoid activation function, and an element-wise product (\otimes) that either activates or shuts down the connection it is associated with. Therefore, the output of each gate can be expressed as follows:

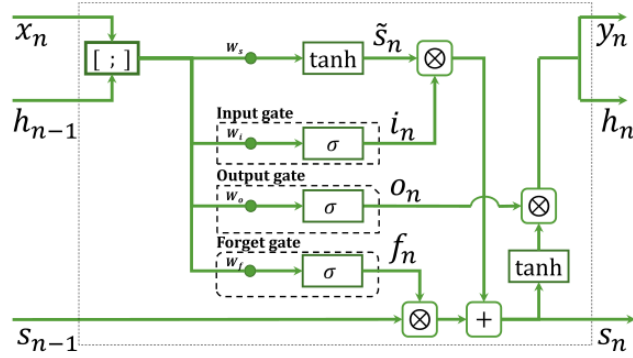


Figure 9 – Most used gated RNN structures. LSTM.

$$f_n = \sigma(W_f \cdot [h_{n-1}; x_n] + b_f) \quad (3)$$

$$o_n = \sigma(W_o \cdot [h_{n-1}; x_n] + b_o) \quad (4)$$

$$i_n = \sigma(W_i \cdot [h_{n-1}; x_n] + b_i) \quad (5)$$

Moreover, the intermediate state \tilde{s}_n and the new cell states s_n and h_n are computed as follows, where $\sigma(\cdot)$ is a sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$.

$$\tilde{s}_n = \tanh(W_s \cdot [h_{n-1}; x_n] + b_s) \quad (6)$$

$$s_n = f_n \otimes s_{n-1} + i_n \otimes \tilde{s}_n \quad (7)$$

$$h_n = o_n \otimes \tanh(s_n) \quad (8)$$

A transformation that involves an affine operation is applied to the input, and this result is further modulated by a hyperbolic tangent activation function, which is controlled by an input gate. The cell state, serving as the memory of the model, is updated using this modified input. The updated cell state is used to compute the output for the subsequent layer, denoted as y_n . To obtain y_n , it is passed through another hyperbolic tangent activation function, which is controlled by the output gate. Finally, both the output and the updated cell state are looped back into the network to ensure the model's recurrent behavior (Bartoli; Marabissi, 2022).

2.2.3 Most Commonly Utilized Loss Functions

In machine learning models, the choice of an appropriate loss function is paramount for precise estimation and evaluation. The consistency in selecting a loss function holds

significance in two crucial aspects. First, within a model, the loss function used for parameter estimation must align with the one employed for model evaluation to guarantee efficient optimization and accurate assessment. Second, when comparing different models, employing consistent loss functions becomes imperative for meaningful comparisons. Discrepancies in this regard can result in erroneous performance evaluations and can add complexity to the decision-making process when determining the superior model (Christoffersen; Jacobs, 2004). A loss function plays a pivotal role in assessing parameter performance during the model-fitting process to the data. It quantifies the disparity between the model's predictions and the desired output (Khan; Wang; Yang, 2016).

One of the most frequently employed loss functions is MSE, which measures the average squared difference between the model's predictions and the actual data points (Khan; Wang; Yang, 2016).

The inclusion of MSE as a fundamental evaluation metric in this proposal is vital for appraising the performance and quality of our predictive model. MSE acts as a quantitative measure to gauge the accuracy and precision of our predictions. The formula (Equation 9) functions by computing the squared differences between predicted values and the actual observations, summing these squared differences across the entire dataset, and subsequently calculating their average. The resulting MSE score offers valuable insights into the extent of disparity between our model's predictions and real-world data (Parera et al., 2019).

$$MSE = \frac{1}{n} \sum_{j=1}^n (Y_i - \hat{Y}_i)^2 \quad (9)$$

A neural network employing the MSE loss function can be utilized to accurately estimate the CQI. This enhances the adaptability of BS in real-time, allowing them to dynamically fine-tune service quality in response to changing channel conditions (Vankayala; Shenoy, 2020). In the context of 5G NR systems, precise estimation of Radio Interface (RI) and CQI plays a vital role in optimizing communication performance. This task is particularly challenging due to factors like delay spread and Doppler spread. To tackle this challenge, a neural network architecture with a single hidden layer and employing the MSE loss function can be employed (Baknina; Kwon, 2020).

In the domain of statistical analysis, the MSE is highly susceptible to outliers. A single outlier with a substantial error can significantly inflate the MSE, making it an inadequate choice for datasets containing such exceptional data points (Raca et al., 2020). Furthermore, it is important to note that the MSE is sensitive to model assumptions, especially the assumption of normally distributed errors with constant variance, as violations of these assumptions can introduce bias into parameter estimates (Fearn, 2012).

Sensitivity to Outliers , The MSE is sensitive to outliers in the data. Large errors (outliers) can significantly skew the loss value. Impact: Outliers disproportionately affect MSE due to the squaring term. A single large error can significantly inflate the loss, mis-

leadingly emphasizing its importance and potentially biasing models towards minimizing extreme errors at the expense of overall accuracy (Roiger, 2017)

MAE, another frequently utilized loss function, offers robustness in measuring error, ensuring a balanced assessment of model performance.

As a loss function, MAE quantifies the degree to which a predictive model's output aligns with the actual observed values. It achieves this by computing the absolute difference between the predicted and actual values for each data point and subsequently averaging these disparities. In essence, MAE assesses the average magnitude of prediction errors (see Equation 10 for the formula).

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (10)$$

Myttenaere et al. (2016) investigate the implications of using the MAPE to assess regression models. It finds that there is an optimal MAPE model and that Empirical Risk Minimization is universally consistent when using MAPE.

Qi et al. (2020) investigate the use of MAE as a loss function for Deep Neural Network (DNN) based vector-to-vector regression. It compares MAE to the commonly used MSE and argues that MAE offers several advantages. Unlike MSE, which assumes Gaussian errors, the paper suggests that MAE can better model errors with a Laplacian distribution. This theoretical argument is then supported by practical experiments, demonstrating that MAE can lead to improved performance in applications like speech enhancement.

The age-old debate regarding Root-Mean-Squared Error (RMSE) and MAE for model assessment persists. While both metrics are widely used, their application isn't always straightforward. Often, both are presented without clear guidance, leaving interpretation to the audience. Recent discussions, like those by Willmott and Matsuura (2005) and Chai and Draxler (2014), delve into the strengths of each metric, often framing it as a binary choice. However, this comparison overlooks the nuances. Neither metric inherently dominates the other. RMSE thrives when errors follow a normal (Gaussian) distribution, while MAE shines with Laplacian distributions. For data deviating from these norms, other metrics might provide a more insightful evaluation. Ultimately, the optimal choice hinges on the specific context and error distribution of your data (Chai; Draxler, 2014).

The RMSE and MAE stand as widely adopted metrics for model evaluation. Despite their prevalence, ongoing confusion surrounds their application, often leading to the convention of presenting both metrics and leaving the determination of relevance to the reader. Recent discussions by Willmott and Matsuura (2005) and Chai and Draxler (2014) revisit the longstanding debate over their preference. However, framing this comparison as a binary choice can be misleading. Neither metric holds inherent superiority: RMSE excels with normal (Gaussian) errors, while MAE shines with Laplacian errors. When errors diverge from these distributions, alternative metrics prove more effective (Hodson, 2022).

2.3 Summary of the Chapter

In this chapter, we provided a concise overview of essential concepts in 5G networks and the prediction of CQI. We then delved into the topics of computer networks, CQI, machine learning, and loss functions in machine learning. This chapter explores how these concepts intertwine to design novel loss functions for CQI forecasting. This approach involves leveraging machine learning techniques within the context of computer networks, specifically focusing on 5G technology, and using appropriate loss functions to optimize the prediction accuracy.

Chapter 3

Related Work

This section outlines the primary studies in the academic realm that center around forecasting parameters in mobile networks through the utilization of deep learning techniques. Notably, there is a particular emphasis on methodologies geared towards predicting CQI within 5G networks. However, the literature also encompasses prediction methodologies tailored for LTE mobile networks, commonly referred to as 4G, as evidenced by Parera et al. (2019), or for various other network parameters, exemplified in the work of Sakib et al. (2020).

Yin et al. (2020) employed LSTM methodology with an online training component to predict the signal quality indicator (CQI) in 5G networks. Their primary goal was to mitigate the impact of outdated CQI values. They initially assessed the consequences of using outdated CQI in 5G systems. Subsequently, through simulating data in the NS-3 environment, the authors introduced and evaluated the effectiveness of their trained proposal. In their approach, the model's predicted CQI value is substituted for the user's CQI value as input to the Medium Access Control (MAC) layer of the eNBs. The authors conducted performance comparisons between their LSTM proposal and CQI prediction algorithms based on Feedforward Neural Networks from the same simulation setup. They explored two distinct models: an LSTM with a fully-connected layer comprising 40 neurons, an intermediate layer with 30 LSTM units, and an output layer summing the intermediate layer's outputs; and a DNN with 40 input neurons, N hidden neurons, and 30 output neurons. Both models receive input windows containing 40 CQI values and use the Scaled Exponential Linear Unit (SELU) activation function to prevent gradient explosion. Individual models are trained for each user and remain active for a specific period, as users exhibit diverse behavior patterns. Consequently, when the system's performance deteriorates rapidly, a new model is generated and trained from scratch to adapt effectively to

the emerging pattern of characteristics.

Vankayala and Shenoy (2020) argue that the demand for CQI by eNBs introduces latency and is not always essential. They aim to categorize when UEs need to request new CQI values to adjust the predictor. In simpler terms, the system only requires new CQI values when they significantly deviate from the actual values. The authors use concepts from information theory and probability to justify their classification for requesting new CQI values. Importantly, they emphasize that having more observations (CQI samples) does not necessarily improve model predictability. For the classifier, the authors employ a streamlined neural network architecture with P i, k inputs. Here, i represents the i -th UE out of I UEs, and k denotes the k -th sub-band of the eNB out of K possible sub-bands. The classifier comprises 3 hidden layers and an output layer with the same dimensionality as the input. If the output is 1, the UE requests its CQI value. However, in their experiments, the authors choose the method of CQI request through stochastic selection. The investigators devised two distinct neural networks for the estimation model. Each network received a vector of CQI values, where absent entries were denoted by 0. Their objective was to forecast the missing values and generate a comprehensive vector of estimated CQIs. The networks varied in the count of hidden layers, one comprising 4 layers while the other comprised 5. Both networks utilized Rectified Linear Unit (ReLU) and were optimized using the Adam optimizer. The model featuring 5 hidden layers, characterized by a more intricate structure and a greater number of parameters, demonstrated superior performance.

Parera et al. (2019) investigate CQI prediction in a 4G network with multiple communication cells operating on various carrier frequencies. Their goal is to achieve this prediction even when little or no information is available for a specific carrier. The authors explore two transfer learning scenarios. In the first, they leverage models trained on different carriers within the same cell. The second scenario involves transferring knowledge from models that operate on the same carrier frequency but in neighboring cells. They begin by evaluating two non-deep learning methods: the mean method (predicting the next CQI as the average of past values) and auto-Auto-Regressive Integrated Moving Average technique (ARIMA). Next, they compare the performance of these methods with a one-dimensional Convolutional neural network (CNN) containing three convolutional layers followed by a fully connected layer. Additionally, they evaluate an LSTM model with an input layer of LSTM units and a final fully connected layer. The models are trained on real-world data from a 4G mobile network. This data consists of hourly averaged CQI measurements, representing the average CQI variation within that time-frame. The time series data is first normalized and then segmented into fixed-size sliding windows of 24 data points.

Numerous challenges in the fields of learning, optimization, and statistics literature (Cesa-Bianchi; Lugosi, 2006; Portnoy; He, 2000) underscore the need for resilient solutions,

mandating that models undergo training or optimization with diminished susceptibility to outliers. This ensures their robustness against outlier influence, in contrast to the impact of inliers, i.e., nominal data (Hastie; Tibshirani; Wainwright, 2015; Huber, 2004). This approach finds widespread application in tasks related to parameter estimation and learning, particularly in cases where prioritizing a robust loss, such as the absolute error, proves more advantageous than opting for a non-robust loss like the quadratic error due to its resistance to substantial errors. When faced with the choice, it becomes essential to transcend traditional outlier detection techniques (Gokcesu et al., 2018; Delibalta et al., 2016), and focus efforts on incorporating inherent resilience to outliers into the design of loss functions. In this context, the Huber loss function emerges as a promising choice, striking a balance between the mean squared error and absolute error, offering robustness to outliers while maintaining sensitivity to inliers. Its adaptive nature makes it well-suited for scenarios where a compromise between the two extremes is crucial for model performance and generalization (Gokcesu; Gokcesu, 2021).

The Huber Loss, as proposed by Jadon et al. (Jadon; Patil; Jadon, 2024), is a robust combination of the MSE and MAE loss functions. It introduces a hyperparameter delta (δ) that determines the transition point between the two functions. When the absolute difference between the predicted and true values is smaller than δ , Huber Loss behaves like MSE. Otherwise, it transitions to a linear MAE-based function. This dual nature helps balance sensitivity to outliers and stability in gradient-based optimization, the Huber Loss function offers the advantage of limiting the impact of outliers through linearity above the δ threshold, unlike MSE.

Adaptive Huber regression is a robust and data-driven solution for handling outliers and heavy-tailed distributions in big data, unlike traditional methods. It automatically adjusts parameters to balance bias and robustness, proven effective across various data scenarios, including those with heavy-tailed distributions (Sun; Zhou; Fan, 2020).

Cavazza and Murino (2016) propose a method for exact Huber loss optimization in scalar regression with semi-supervised learning. Their approach incorporates multi-view learning, which leverages information from multiple data perspectives, and manifold regularization. Additionally, they employ a data-driven adaptation of the Huber loss threshold and actively balance the use of labeled data to mitigate the impact of noisy or inconsistent annotations during training.

Considering these works, Table 1 presents a summary of the main characteristics of each one. It can be highlighted that the proposal of this work differs from the others by the proposed RAHL. This function allows the model to learn the optimal level of robustness during training. By optimizing this unique loss function, we navigate the delicate balance between precision and outlier resilience, minimizing errors and ensuring heightened prediction accuracy.

Work	5G	CQI	LSTM	CNN-1D	Huber	Adaptive Huber
(Parera et al., 2019)	✓	✓				
(Yin et al., 2020)	✓	✓	✓			
(Vankayala; Shenoy, 2020)	✓	✓		✓		
(Gokcesu; Gokcesu, 2021)					✓	
(Sun; Zhou; Fan, 2020)						✓
(Cavazza; Murino, 2016)					✓	
(Jadon; Patil; Jadon, 2024)					✓	

Table 1 – Summary listing the main characteristics of the works cited.

3.1 Summary of the Chapter

In this chapter, we provided an extensive overview of the latest advancements in 5G Networking, including insights into channel quality, models, and the application of loss functions. We have delineated the principal characteristics of the relevant works and described how our approach extends beyond the existing research.

Chapter 4

Using Residual-based Adaptive Huber Loss for prediction of CQI

In this chapter, we delineate the profound framework crafted within this study, centered on the prediction of forthcoming CQI values utilizing the RAHL loss function within 5G networks. The overarching framework is depicted in Figures 10 and Figure 11. Figure 10 outlines the anticipated scenario for implementing the framework, presuming a UE (static or vehicular) tethered to a 5G network, periodically transmitting CQI values via control signals to the eNB. These values are then relayed by eNB stations to the 5G cloud core, where a historical repository of these values is maintained in the form of time series. Subsequently Figure 11, these historical records undergo preprocessing and serve as the basis for training deep learning models, which are subsequently extrapolated to infer future CQI values based on new data received at the 5G core. However, given the unavailability of a genuine 5G cloud infrastructure, actual transmission histories sourced from three real dataset (Mustafa, 2023), (Raca et al., 2020, Available: <https://doi.org/10.1145/3339825.3394938>), (Intrig-unicamp, 2023) were utilized to simulate the CQI transmission phase of the framework.

4.1 Data set

We investigate three datasets:

□ Dataset A:

A dataset comprises Channel Level Metrics (CLM) files and YouTube QoE logs stored in MySQL, featuring metrics like Timestamp, Location, Network details,

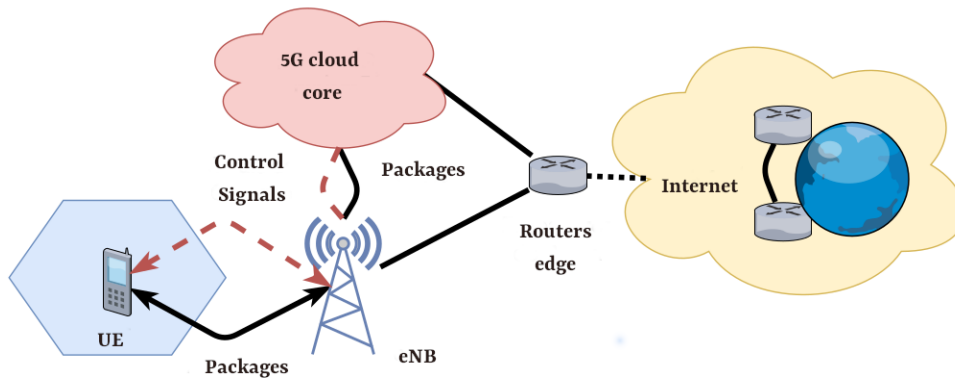


Figure 10 – General structure adopted to execute the CQI prediction framework in 5G networks.

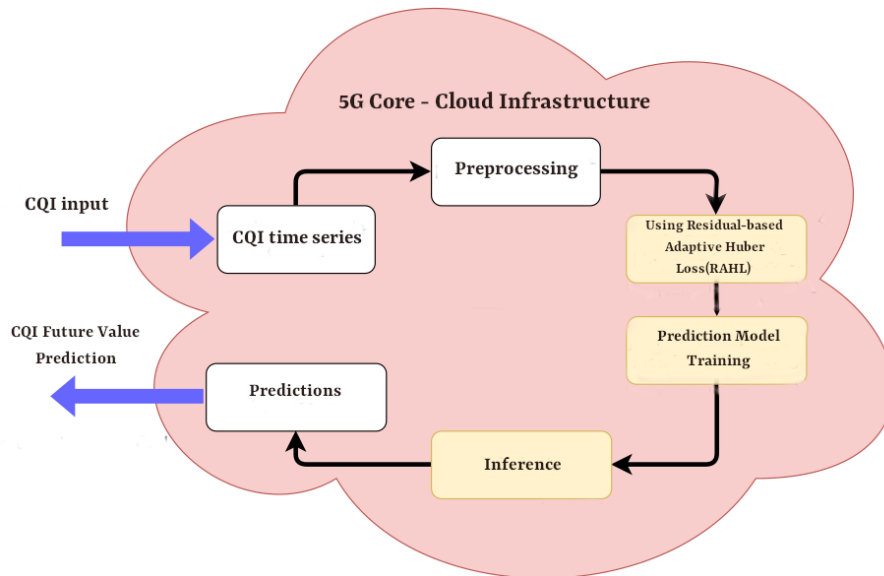


Figure 11 – General structure adopted to execute the CQI prediction framework in 5G networks.

Signal Strength, Bitrates, Altitude, and Experiment ID (EID). The data, obtained from a comprehensive 5G collection campaign in diverse scenarios (Mobility, Pedestrian, Indoor, Outdoor), utilized the YouTube IFRAME API and Android's Network Monitor app. The dataset captures a wide range of parameters, providing insights into 5G network performance across various use cases (Mustafa, 2023).

□ Dataset B:

The dataset comprises 83 records of Internet transmissions recorded by G-NetTrack v18.7 on a Samsung S10 connected to an Irish mobile operator. It includes 3142 minutes of transmission logs, organized into three services (File Download, Amazon Prime, and Netflix) and two mobility patterns (Static and Vehicular). The logs, limited by an 80GB data plan, are stored in a CSV file with fixed features and variable data points. The dataset provides detailed attributes such as timestamp, geographi-

cal coordinates, node velocity, mobile operator (anonymized), cell ID, network mode, bitrates, device state, and various signal quality indicators for both the primary and neighboring cells (Raca et al., 2020, Available: <https://doi.org/10.1145/3339825.3394938>).

□ Dataset C:

Field tests in Brazil through a 5G network using a Samsung S21 5G, focusing on traffic and mobile network monitoring. YouTube metrics captured through various clients were analyzed alongside manual monitoring using G-NetTrack Pro in 5G-covered areas in São Paulo. Data was enriched with Anatel’s Mosaico information on registered telecommunication stations, providing details on technologies, equipment, frequencies, locations, licensing, and ownership (Intrig-unicamp, 2023).

4.2 The deep learning-based CQI prediction model

Building on previous work, we employ the well-known LSTM model due to its ability to capture long-term dependencies in sequential data, which makes it ideal for accurately predicting CQI values. It is not our intention to go deeper in the LSTM architecture but shortly introduce how it works.

Figure 12 - Source: Adapted from (Bartoli; Marabissi, 2022) illustrates the LSTM architecture, designed to handle input sequences with one-dimensional features. The sequence first passes through an LSTM layer with 64 hidden units, enabling it to learn sequential patterns. Next, a fully connected layer with 64 units introduces non-linear transformations to the LSTM output. Finally, the regression layer generates a single output value, representing the model’s prediction for the given sequence (Bartoli; Marabissi, 2022).

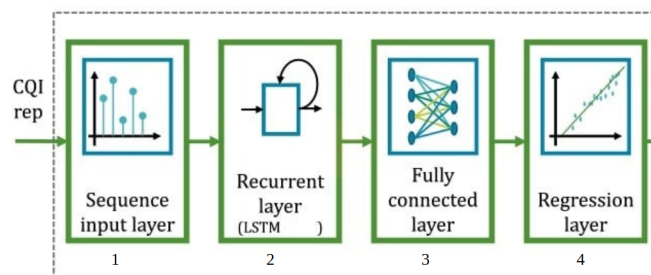


Figure 12 – Considered LSTM scheme.

4.3 Huber Loss Function

The Huber loss is a widely adopted loss function in regression tasks, especially in situations where the data may include outliers or noise. It strikes a harmonious balance between the MSE (quadratic loss) and the MAE (absolute value loss), rendering it more

resilient to extreme data points compared to relying solely on mean squared error. The transition point within the Huber loss determines when it transitions from a quadratic to an absolute value behavior, making it a crucial hyperparameter that significantly impacts the performance of a regression model. Nevertheless, selecting the optimal transition parameter, which governs the shift between quadratic and absolute value behavior, has historically presented a challenge, necessitating frequent hyperparameter searches to find the best value (Meyer, 2021).

In the domain of regression, two commonly used loss functions are the L1 (Equation 11) and L2 (Equation 12) loss functions, identified as:

$$L1(x) = |x| \quad (11)$$

and

$$L2(x) = 1/2x^2 \quad (12)$$

Each of these functions has distinct advantages and drawbacks. L1 is known for its robustness in the presence of outliers within the dataset but lacks differentiability at zero. In contrast, L2 is differentiable across its entire domain but tends to be highly sensitive to outliers.

Huber introduced an alternative loss function known as the Huber loss (Equation 13), defined as follows:

$$H(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{if } |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{if } |y - f(x)| > \delta \end{cases} \quad (13)$$

Here, $\delta \in \mathbb{R}^+$ represents a positive real number that controls the transition from L1 to L2. The Huber loss displays both differentiability across its entire range and robustness to outliers, This hyper parameter (δ) usually needs to be manually chosen.

4.4 Residual-based Adaptive Huber Loss (RAHL) - the improved Huber loss for CQI prediction

In the realm of regression problems, the absolute loss, $L_1(y, f_\theta(x)) = |y - f_\theta(x)|$, and the ubiquitous quadratic (squared) loss, $L_2(y, f_\theta(x)) = (y - f_\theta(x))^2$, emerge as consequential alternatives, requiring a strategic choice rooted in their distinctive characteristics. The quadratic loss, recognized for its robust convexity, facilitates rapid learning rates, while the absolute loss is esteemed for its inherent resilience. This dichotomy emphasizes the need of amalgamating the strengths of both loss functions, leading to models that not only exhibit robustness against outliers but also achieve swift convergence with optimal goodness of fit.

The Huber loss emerges as a prevalent solution, seamlessly combining quadratic and absolute losses to formulate a resilient loss function that converges quickly (Gokcesu et al., 2018). The Huber loss is widely adopted in regression tasks, especially when dealing with outliers or noise in the data. It achieves a balanced compromise between MSE (quadratic loss) and MAE (absolute loss), offering improved resilience against extreme data points compared to relying solely on MSE or MAE. The pivotal transition point within the Huber loss dictates its shift from quadratic to absolute loss behavior, making it a crucial hyperparameter that significantly influences the performance of a regression model. Nevertheless, the challenge lies in selecting the optimal transition parameter, necessitating frequent hyperparameter searches to identify the most suitable value (Meyer, 2021).

Formally, the Huber loss is given by (Equation 14):

$$H(y, f_{\theta}(x)) = \begin{cases} \frac{1}{2}(y - f_{\theta}(x))^2, & \text{if } |y - f_{\theta}(x)| \leq \delta \\ \delta|y - f_{\theta}(x)| - \frac{1}{2}\delta^2, & \text{if } |y - f_{\theta}(x)| > \delta \end{cases}, \quad (14)$$

where y is the ground truth, $f_{\theta}(x)$ is a model defined by the learnable parameters θ , and δ is a positive hyperparameter that acts as a pivotal regulator, transitioning the penalty from L_2 to L_1 . This crucial hyperparameter balances the critical trade-off between model accuracy and robustness to outliers. Choosing the right value for δ is crucial, as it determines the transition point where the loss function switches from prioritizing precision to emphasizing robustness. A smaller δ favors L_1 behavior, boosting accuracy but decreasing resilience to outliers. Conversely, a larger δ pushes the loss function toward L_2 characteristics, enhancing robustness but potentially at the cost of accuracy.

Tuning the hyperparameter δ by hand poses substantial challenges for adopting the Huber loss to train regression models, which include subjectivity, data dependence, computational burden, potential overfitting, and the difficulty of balancing accuracy and robustness. This requires a cautious approach and exploration of more sophisticated hyperparameter tuning methods for robust regression models. To mitigate these shortcomings, we propose RAHL, a transformative approach that empowers the model to automatically determine the optimal penalty scheme.

Figure 13 compares RAHL and the other loss functions, illustrating how outliers affect the solution (i.e., model). In this figure, outliers are represented by red points and inliers by blue points. Solving regression problems with MSE penalty results in a model (purple line) that heavily leans towards outliers. Conversely, employing MAE yields a model (red line) close to inliers, neglecting outliers (red points). Introducing the Huber loss allows manual tuning of the hyperparameter δ , where a large δ mimics MSE (black line) and a small δ mimics MAE (brown line). Consequently, carefully selecting this hyperparameter is vital, as it directly affects the model's sensitivity to outliers and, ultimately, its performance in robust regression tasks. The underlying idea of RAHL is to find a balance,

offering sensitivity to small errors (blue points) while maintaining robustness to outliers (red points), as depicted by the green line—ultimately providing the most optimal model for these data.

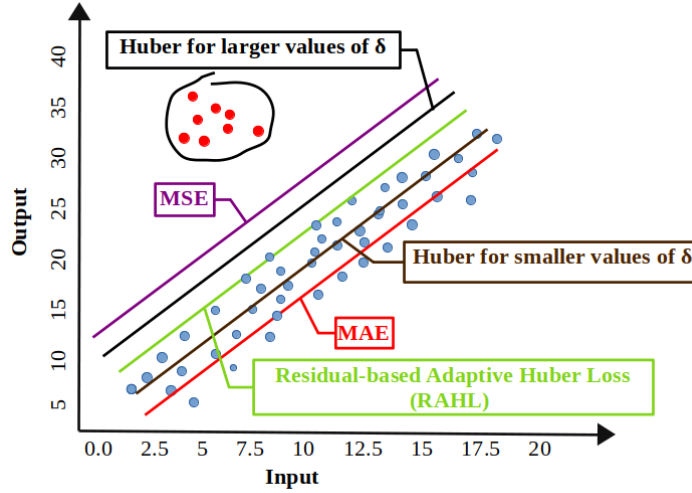


Figure 13 – Exploring how different loss functions shape regression models.

Mathematically, RAHL is identical to the Huber loss, but instead of using a fixed value for the hyperparameter δ , it is computed by (Equation 16):

$$RAHL(y, f_{\theta}(x)) = \begin{cases} \frac{1}{2}(y - f_{\theta}(x))^2, & \text{if } |y - f_{\theta}(x)| \leq \delta \\ \delta|y - f_{\theta}(x)| - \frac{1}{2}\delta^2, & \text{if } |y - f_{\theta}(x)| > \delta \end{cases}, \quad (15)$$

$$\delta = \alpha + \text{ELU}(\beta), \quad (16)$$

where α is a positive hyperparameter defining the initial value for δ and β is a learnable parameter that is added to α . To bound the output and obtain a positive value for δ , the Exponential Linear Unit (ELU) (Clevert; Unterthiner; Hochreiter, 2016) function is applied to the parameter β . ELU is an activation function that performs the identity operation on positive inputs and an exponential non-linearity on negative inputs and is given by (Equation 17):

$$\text{ELU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha(e^x - 1), & \text{if } x < 0 \end{cases}, \quad (17)$$

where α is a constant that defines function smoothness when inputs are negative and is usually set to 1.0. By setting this constant with the same value chosen for the hyperparameter α from Equation 16 (i.e., the initial value for δ), we constraint its output value to the range $[-\alpha, +\infty)$, ensuring that δ is always positive.

Figure 14 illustrates how changing δ impacts the Huber loss, as compared to absolute and quadratic losses. Each line depicts the loss (y-axis) as a function of the residual (x-axis), which is given by the difference between the groundtruth value (y) and the model's

prediction (\hat{y}). The larger the δ , the more the Huber loss behaves like MSE, conversely, the smaller the δ , the more the Huber loss behaves like MAE. The key advantage of RAHL is to enable the model to adapt based on the distribution of errors, learning to behave more similar to MSE or MAE based on the data.

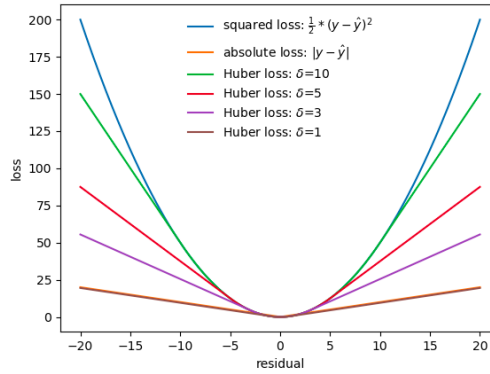


Figure 14 – The Huber loss for various values of δ , moving between MSE and MAE.

4.5 Predictions of future values

For LSTM, since its forwarding process happens through a recurrence loop, where each step of the recurrence generates an output, the model's prediction is defined as the output of its last recurrence block.

Thus, in a real-world scenario, once the model starts generating degraded CQI values, this information can be sent in advance to the eNBs so that they can take corrective actions to meet the expected requirements for 5G connections.

Chapter 5

Experimental Evaluation

In this chapter, the experimental protocol used for evaluating the proposed framework and the results obtained from executing the experimental protocol are described. Initially, the experimental procedure is detailed, outlining the preprocessing steps of transmission records, the training strategy of deep models, and the definition of the hyperparameters used. Finally, the obtained results are presented and discussed.

5.1 Experimental procedure

The success of deep learning models hinges on their ability to learn from data. To achieve this, we follow a detailed process. First, data is preprocessed to ensure the model can understand it. Then, the model is trained from scratch using a designated training set, with various loss functions guiding the learning process. Finally, its performance is thoroughly evaluated on a separate validation set. By analyzing results across experiments, we can identify the most effective loss function for a specific dataset.

5.1.1 Data preparation

We tackle missing data challenges by leveraging Not a Number (NaN) for placeholder values, a standard practice in Python's numerical domain. Data normalization is an efficient method to enhance the accuracy of specific machine learning models, and some machine learning models do not perform optimally without it. In our proposed model, we employed Min-Max Scaler (MMS) as the normalization technique. MMS essentially rescales the data within the range of $[0, 1]$ or $[-1, 1]$. The mathematical formula for MMS is represented as follows (Equation 18) (Dey; Hossain; Rahman, 2018):

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (18)$$

With MMS we achieve uniformity and improve model performance.

Also the records are filtered only at points connected to the 5G network. Filtering at 5G points, for each record, generates a set of contiguous point intervals, where each interval is not adjacent to the others, and, for example, if these different intervals were concatenated again, discontinuities would be inserted into the resulting time series. Because of this, each of the contiguous point intervals (provided they have a minimum amount of points) is treated as independent datasets, and time series data of the CQI is extracted for each of these independent point intervals. An example of this procedure is shown in Figure 15 - Source: Adapted from (Silva, 2022), where an illustrative CQI time series is divided into contiguous intervals of points connected to the 5G network.

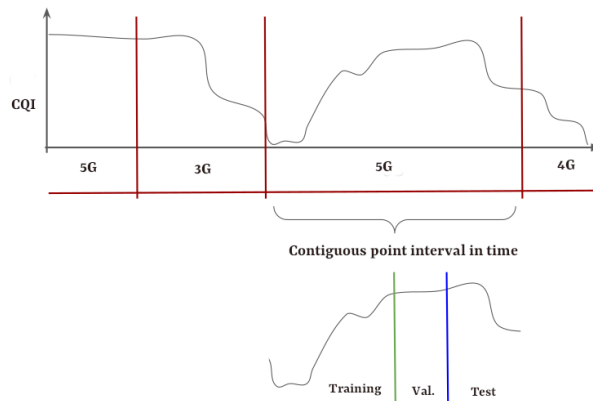
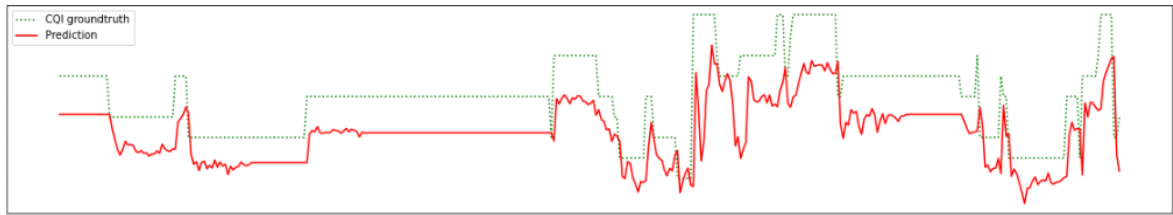


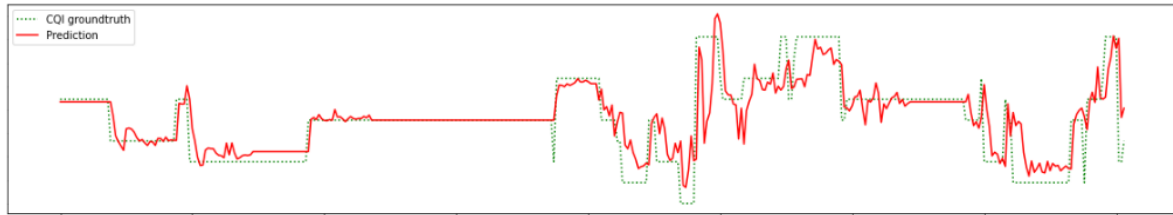
Figure 15 – Splitting a transmission record.

After obtaining the time series, the points are first divided into a training set, comprising the first 80% of the points, and a test set, comprising the last 20% (Figure 15). As we mentioned, both the training and test sets are normalized to values within the range of -1 to 1 using the minimum and maximum values calculated only from the training set's time series. Normalization is necessary because, as shown in Figure 16 - Source: Adapted from (Silva, 2022) with the results obtained by the LSTM model for CQI prediction (discrete values from 0 to 15) with and without normalized data, it was found that not normalizing the data hinders the model's learning, usually leading to a fixed offset in the predicted values compared to the expected values.

Subsequently, a sliding window of size w with a step of 1 is iterated over the normalized time series of the training and test sets, ultimately obtaining a set of windows for each dataset. All windows are then associated with a label, which in this case refers to the future CQI value immediately following the end of the current window. Since each contiguous interval is treated as an independent dataset, the total number of observations for each interval varies significantly, Figure 17 - Source: Adapted from (Parera et al., 2019) shows this procedure in detail (Parera et al., 2019).



(a) Predictions with a model trained on unnormalized data.



(b) Predictions with a model trained on normalized data.

Figure 16 – Inference with models trained on normalized and denormalized data.

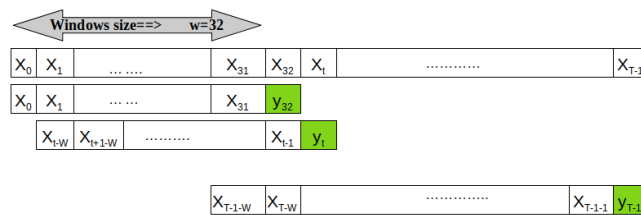


Figure 17 – Sliding windows for time series forecasting.

Finally, now that the data is properly adjusted for the input dimensionality of the deep model, the training set is once again divided into a (new) training set (80% of the training data) and a validation set (the final 20% of the training data) as a good practice to avoid evaluating the model directly on the test set during the training phase.

5.2 Training and tuning the parameters of ML

Once the data is available in its correct dimensionality, individual deep model is constructed for each of the contiguous time window sets and trained from scratch on their training subset. The training process on the training data is repeated for several epochs, where for each epoch, the model performs a complete pass over all the data in the training subset, executing its data forwarding procedures to generate predictions and calculate the loss function, and performing backpropagation to adjust its internal weights towards minimizing the loss function. In each epoch, after the training step (forwarding and backpropagation), the model is evaluated on the validation dataset. In particular, machine learning models are generally directed towards objective functions, which translate the problem into a mathematical formula that will be minimized or maximized.

Since the proposed framework is designed for predicting future values, we apply various

loss functions—MSE, MAE, and Huber loss—to compare their performance against the results obtained using the RAHL loss function.

We implemented the Python code for our project and executed it using Google Colab, a cloud-based platform for collaborative coding and data analysis. Also, we defined the network architecture and training hyperparameters for the LSTM model, which are summarized in Figure 18.

Input Size	1	Number of Epochs	300
Hidden Size	64	Mini-Batch Size	24
Number of Stacked Layers	1	Windows Size	36
Number of FC Layers	1	Initial Learning Rate	0.01
FC Hidden Size	64	Optimizer	Adam
Activation Function for FC Layers	ReLU		
Output Size	1		

(a) Network architecture
(b) Training hyperparameters

Figure 18 – Network architecture and training hyperparameters for the LSTM model.

5.3 Performance metrics

To ensure a fair comparison of the model performance across different loss functions, we utilize the MAPE . Our approach involves training the model with different loss functions, and using MAPE as the primary validation and testing metric. Widely used in statistics and data analysis, particularly for time series forecasting, MAPE expresses error as a percentage, as shown in Equation 19, where lower values indicate better performance (Myttenaere et al., 2015).

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \times 100\% \quad (19)$$

5.4 Results

Initially, we trained the LSTM model using the Huber loss and manually selected the hyperparameter δ . To do so, we tested various values for δ , ranging from 0.5 to 4.0 by a step of 0.5. The results obtained for each dataset are presented in Table 2 and show the performance variations across diverse values for δ and distinct categories within datasets, highlighting the lowest MAPE values for each setting. In this table, MAPE can be interpreted as a measure of forecast performance: lower MAPE values indicate more accurate forecasts, while higher MAPE values indicate less accurate forecasts. After extensive computations across various values for δ , the best choice was identified as the minimum error. However, it is crucial to note that this minimum may not be the optimal

solution, as further analysis will demonstrate. As expected, there is no silver bullet for all cases: the performance for the Huber loss often depends on the choice for the hyperparameter δ .

Table 2 – MAPE values for the Huber loss with different values for δ .

Dataset	$\delta = 0.5$	$\delta = 1$	$\delta = 1.5$	$\delta = 2$	$\delta = 2.5$	$\delta = 3$	$\delta = 3.5$	$\delta = 4$
A-Indoor	7.55	7.56	6.89	7.42	7.72	6.33	7.55	7.10
A-Pedestrian	26.05	25.79	26.81	27.66	27.34	27.57	25.83	26.38
A-Mobility	18.49	18.77	19.35	16.67	17.62	19.11	15.74	15.63
A-Outdoor	72.24	59.05	54.13	61.22	70.31	52.74	58.13	56.88
B-Static-Netflix	14.61	14.29	15.46	18.05	16.35	11.93	11.19	12.87
B-Static-Download	18.74	13.87	15.93	14.07	13.49	16.51	16.67	15.94
B-Static-Amazon_Prime	1.089	1.023	1.1	1.21	1.26	1.19	1.12	1.11
B-Driving-Netflix	62.09	67.65	66.97	58.08	57.6	60.26	62.5	66.66
B-Driving-Download	41.31	39.53	46.15	40.82	45.75	46.87	45.29	38.99
B-Driving-Amazon_Prime	20.3	28.44	24.02	23.18	24.04	18.81	19.61	19.08
C	25.59	29.25	25.48	23.93	23.85	23.08	22.67	24.2

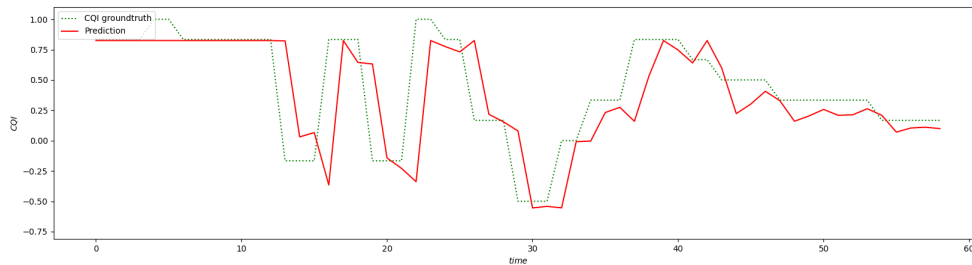
Table 3 compares the results for RAHL with those obtained for the Huber loss, considering the best δ values found for each setting, according to the previous experiment. Unlike the Huber loss, in RAHL the hyperparameter δ is transformed into a trainable parameter, eliminating the need for choosing δ by hand. The results for MSE and MAE were also included for comparison. Consistently across all the datasets, the MAPE values were lower when using RAHL, indicating its superior performance relative to the other loss functions examined.

To highlight the benefits of performing CQI prediction using a model trained with RAHL, we compare the ground truth value and the model’s prediction for some samples from our datasets. For this analysis, we took samples from three distinct datasets. The first and second respectively, taken from the Mobility category of the dataset A and the Amazon_Prime of the dataset B, comprises a relatively small samples with significant

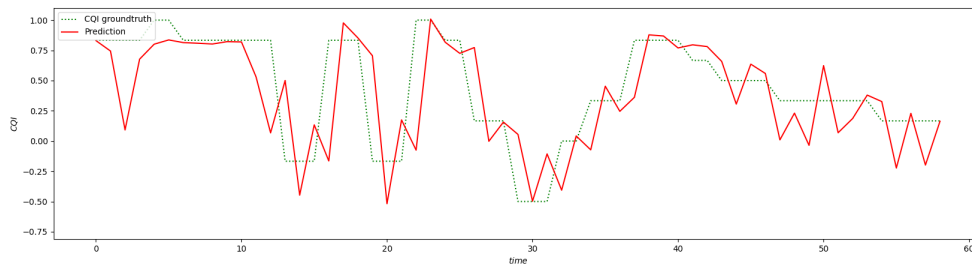
Table 3 – MAPE values obtained for different loss functions.

Dataset	RAHL	Huber loss (best δ)	MSE	MAE
A-Indoor	5.77	6.33	7.07	5.80
A-Pedestrian	21.98	25.79	28.44	23.66
A-Mobility	13.27	15.63	17.35	13.5
A-Outdoor	49.48	52.74	63.44	55.16
B-Static-Netflix	7.8	11.19	12.6	9.34
B-Static-Download	11	13.49	13.53	12.34
B-Static-Amazon_Prime	0.82	1.023	1.27	0.95
B-Driving-Netflix	54.1	57.6	70.13	60.27
B-Driving-Download	36.8	38.99	45.71	38.32
B-Driving-Amazon_Prime	17.8	18.81	19.31	18.66
C	21.99	22.67	26.71	23.11

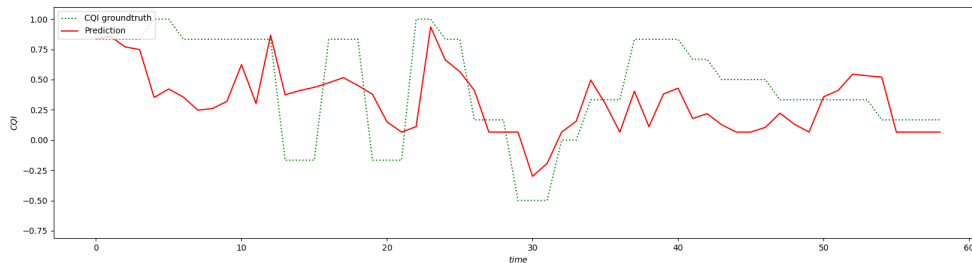
outliers. The third sample, selected from dataset C, was larger, had a more uniform distribution, and exhibited fewer outliers.- Figures 19 , 20 and 21 compare, for each of the chosen samples, respectively, the actual CQI values (green line) and the predictions (red line) made by LSTM models trained with different loss functions.



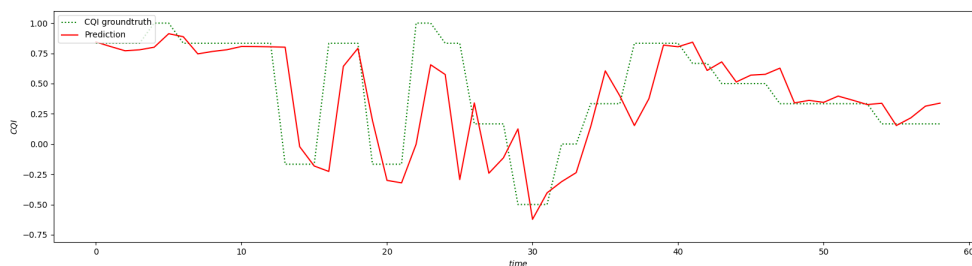
(a) Residual-based Adaptive Huber Loss (RAHL) - MAPE=6.55.



(b) Huber loss (best δ) - MAPE=9.05.



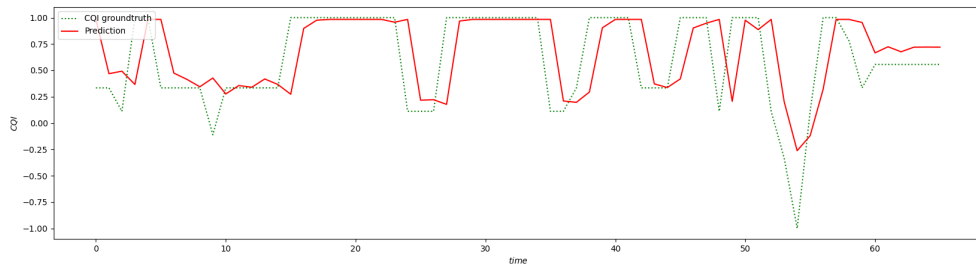
(c) Mean Square Error (MSE) - MAPE=10.37.



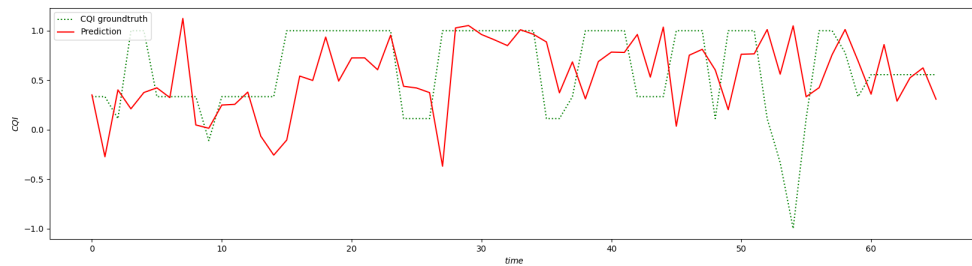
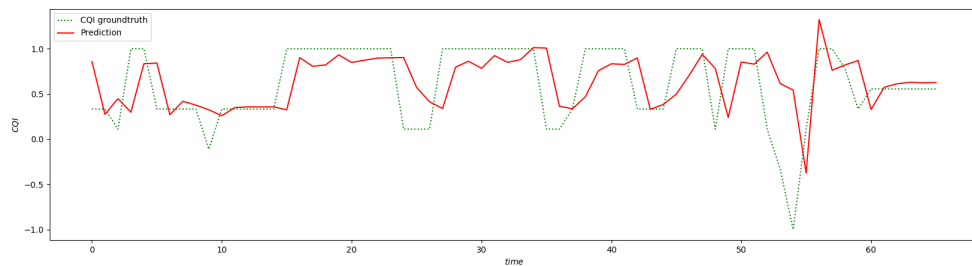
(d) Mean Absolute Error (MAE) - MAPE=6.73.

Figure 19 – Comparison of the ground truth value and the model's prediction for a sample from the Mobility category of the dataset A.

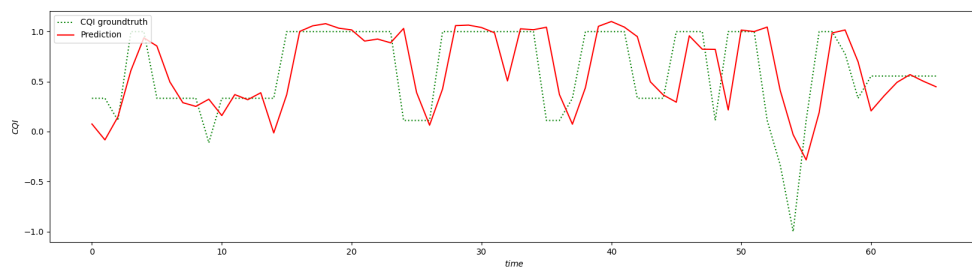
As the subcaptions of each figure reveal, the MAPE value for RAHL is the lowest



(a) Residual-based Adaptive Huber Loss (RAHL) - MAPE=9.90.

(b) Huber loss (best δ) - MAPE=15.07.

(c) Mean Square Error (MSE) - MAPE=11.80.

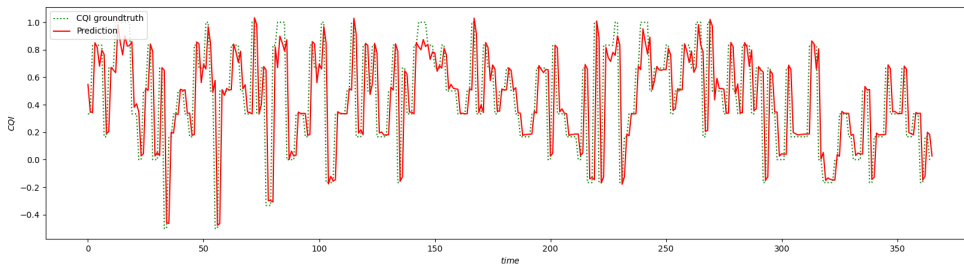


(d) Mean Absolute Error (MAE) - MAPE=10.60.

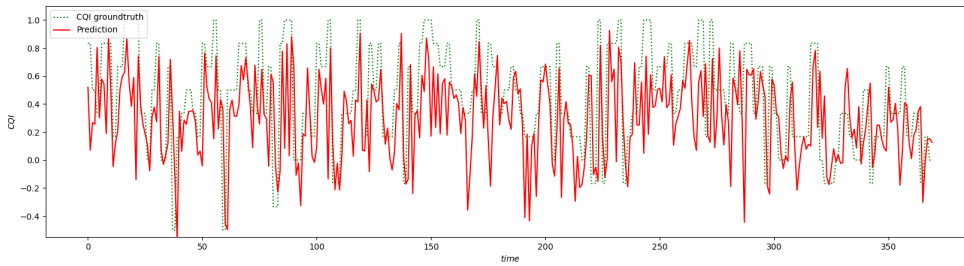
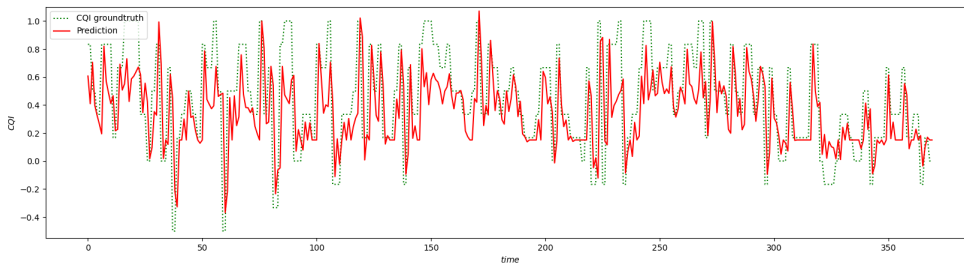
Figure 20 – Comparison of the ground truth value and the model's prediction for a sample from the Amazon_Prime category of the dataset B.

in both samples. Additionally, the predictions for the LSTM model trained with RAHL consistently follow the actual CQI values more closely, regardless of sample size (time duration). This indicates the robustness of RAHL against outliers and overall superiority in achieving accurate results for CQI forecasting.

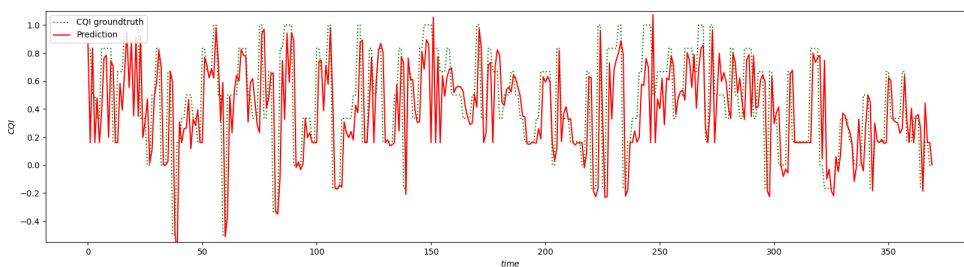
It is of paramount importance to remember that CQI is a metric for quantifying the



(a) Residual-based Adaptive Huber Loss (RAHL) - MAPE=11.07.

(b) Huber loss (best δ) - MAPE=13.64.

(c) Mean Square Error (MSE) - MAPE=14.20.



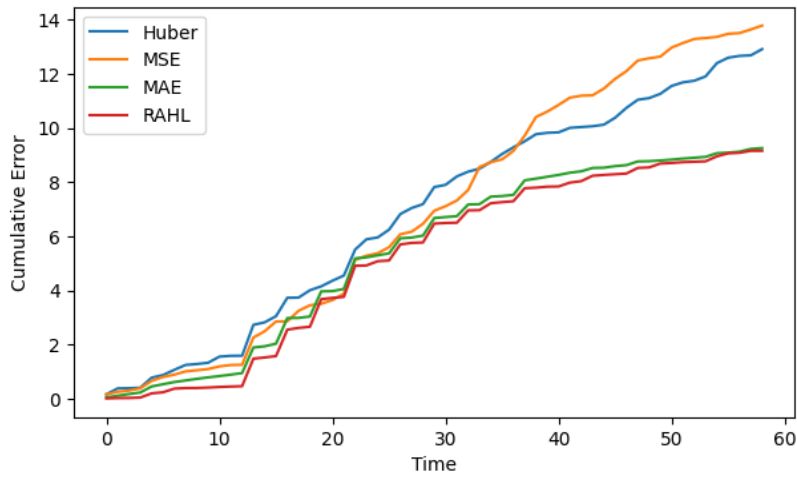
(d) Mean Absolute Error (MAE) - MAPE=12.16.

Figure 21 – Comparison of the ground truth value and the model’s prediction for a sample from the dataset C.

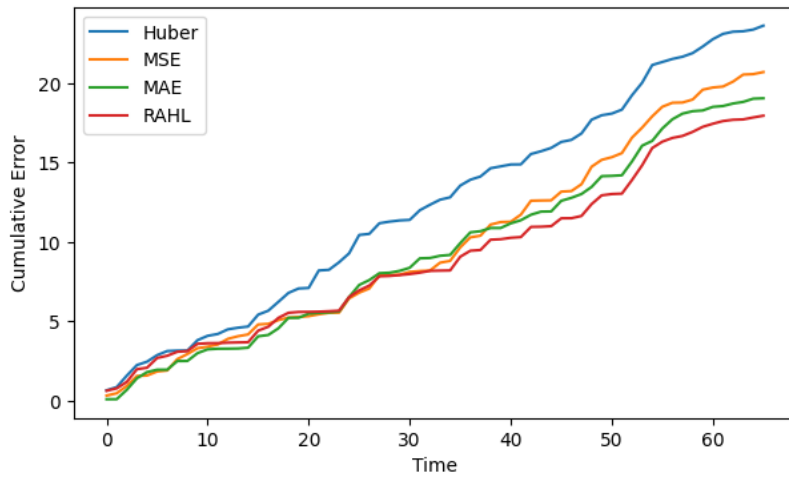
quality of the radio channel between the UE and the base station. The CQI enables the base station to dynamically adapt the modulation for each UE so that the data rate can be optimized. As a consequence, making a wrong CQI prediction will affect negatively how the modulation is configured and how the resource allocation in the 5G network will be done. As an example, by analyzing the predictions made by the LSTM model trained

with Huber loss (best δ) in Figure 19(b), we can observe from time 60 that the prediction is (wrongly) going up and down, suggesting some network issue. However, the actual CQI values indicate that the network connection is stable for most of the time. The same can be observed in Figure 19(c) in which the CQI prediction is wrong during almost the entire period.

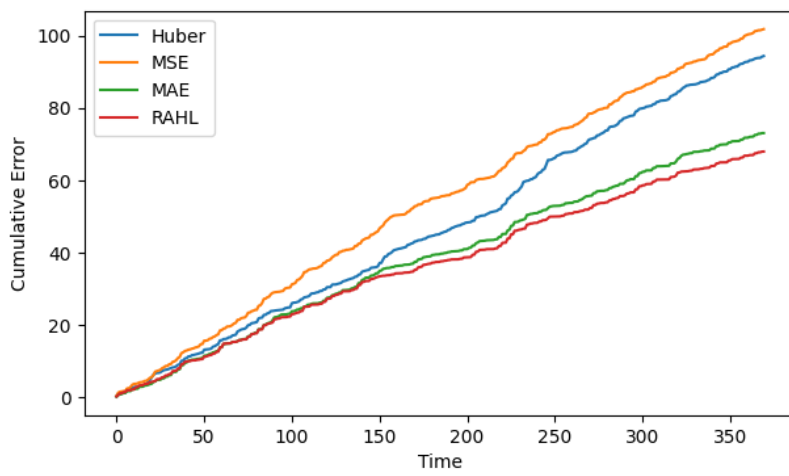
Figure 22 presents a different view of such results, showing how the absolute percentage error is accumulated over time. In this way, we can analyze the error patterns for LSTM models trained with different loss functions. As expected, the cumulative error associated with RAHL grows slower than that of other loss functions and, for this reason, it is demonstrably more effective for CQI prediction.



(a) Dataset A-Mobility.



(b) Dataset B-Amazon_Prime.



(c) Dataset C.

Figure 22 – Absolute percentage error accumulated over time for different losses.

Chapter 6

Conclusion

The CQI is an essential metric for representing the quality and efficiency of the 5G communication channel. It plays a crucial role in guiding the base station's decisions regarding resource allocation, modulation schemes, and coding strategies. These decisions are fundamental to optimizing the network's data rate and capacity utilization, which in turn have a direct impact on the end user's QoE. Accurate CQI predictions are vital for maintaining high-performance 5G networks, where even minor inaccuracies can lead to significant performance degradation and a diminished user experience.

In this study, we explored the use of ML models to predict the CQI in 5G networks. Accurate CQI predictions are crucial for enhancing network performance, as they directly impact resource allocation and the quality of service users experience. Typically, loss functions like MSE, MAE, and Huber Loss are utilized to reduce prediction errors during ML model training. However, these loss functions face certain challenges when applied in dynamic and complex environments like 5G networks.

A significant challenge is their sensitivity to outliers, which can distort the error metrics and negatively influence the model's performance. Additionally, finding the right balance between accuracy and robustness in predictions becomes difficult, particularly in datasets with variability or noise. Moreover, achieving optimal performance with these loss functions often necessitates meticulous manual tuning of hyperparameters, which can be a demanding and time-intensive process, especially in environments like 5G networks where data characteristics frequently change.

To address these challenges, we introduced the RAHL, which represents a significant advancement in the field of CQI prediction. In this study, we have developed RAHL as an enhancement of the Huber loss function, which this method try to learn the hyperparameter delta itself, RAHL tries to learn a residual that is added to it. In this way, we easy

the training of deep models, leading to improved performance. Extensive experimental results, as shown in Figures 19, 20, 21 and 22, consistently indicate that RAHL surpasses the other mentioned loss functions, resulting in lower error rates and more accurate CQI predictions. This improvement is especially important in practical applications where precise CQI predictions are critical for optimizing network performance and enhancing user satisfaction.

One of the key advantages of RAHL is its ability to eliminate the need for manual hyperparameter tuning, a process that is often subjective, time-consuming, and prone to errors. By automating this aspect of the model training process, RAHL reduces the risk of overfitting and enhances the robustness of the model, making it more adaptable to varying data conditions. This capability is especially valuable in 5G networks, where the data environment is constantly changing, and the ability to quickly and accurately adapt is essential. Our findings offer valuable insights for both practitioners and researchers in wireless communications and machine learning. For researchers, RAHL presents new opportunities for further study, particularly in the design of adaptive loss functions capable of dynamically adjusting to enhance performance across various tasks and environments.

However, while our study has made significant progress, we acknowledge that it is not without its limitations. For instance, RAHL's effectiveness has been primarily tested on a specific type of time series dataset related to 5G CQI prediction. To fully understand its generalizability, future research should focus on applying RAHL to a broader range of datasets, including those with different temporal characteristics, noise levels, and distributions. For example, researchers could examine its performance on financial time series, climate data, or healthcare-related datasets, each of which presents unique challenges in terms of seasonality, trend components, and outlier management. Additionally, it would be valuable to explore RAHL's applicability to other features of 5G technology, such as network traffic prediction, user mobility patterns, or resource allocation optimization. This would provide a more comprehensive assessment of RAHL's strengths and weaknesses across various domains and use cases. It is important to investigate how RAHL can be combined with various machine learning models and neural network architectures, including LSTM networks, CNNs, and Transformer models. Each architecture has unique strengths: LSTM networks excel at capturing long-term dependencies, CNNs are effective at identifying spatial patterns, and Transformers utilize attention mechanisms to manage sequential data. Integrating RAHL with these models may enhance their predictive accuracy and robustness for diverse applications. Additionally, with the rapidly evolving landscape of 5G networks, ongoing research is necessary to develop adaptive methods that can continuously improve CQI prediction accuracy in real-time. This may involve exploring online learning methods, reinforcement learning, or hybrid approaches that merge RAHL with traditional error-correction techniques, leading to systems that are both responsive and resilient to changing network conditions.

In summary, the introduction of RAHL represents a significant contribution to the field of CQI prediction, offering a robust, adaptive, and efficient solution for minimizing error rates. Its innovative approach not only improves prediction accuracy but also simplifies the process of hyperparameter selection, making it a valuable addition to the machine learning toolkit. The advancements presented in this study have the potential to greatly enhance the performance of 5G networks, ensuring that they can meet the growing demands of users and continue to deliver high-quality service in an increasingly connected world.

Bibliography

- Alth e, F.; Fortelle, A. de L. An lstm network for highway trajectory prediction. In: IEEE. **2017 IEEE 20th international conference on intelligent transportation systems (ITSC)**. 2017. p. 353–359.
- Baknina, A.; Kwon, H. Adaptive cqi and ri estimation for 5g nr: A shallow reinforcement learning approach. In: IEEE. **GLOBECOM 2020-2020 IEEE Global Communications Conference**. 2020. p. 1–6.
- Balieiro, A.; Dias, K.; Guarda, P. A machine learning approach for cqi feedback delay in 5g and beyond 5g networks. In: IEEE. **2021 30th Wireless and Optical Communications Conference (WOCC)**. 2021. p. 26–30.
- Bartoli, G.; Marabissi, D. Cqi prediction through recurrent neural network for uav control information exchange under urlc regime. **IEEE Transactions on Vehicular Technology**, IEEE, v. 71, n. 5, p. 5101–5110, 2022.
- Cavazza, J.; Murino, V. Active regression with adaptive huber loss. **arXiv preprint arXiv:1606.01568**, 2016.
- Cesa-Bianchi, N.; Lugosi, G. **Prediction, learning, and games**. : Cambridge university press, 2006.
- Chai, T.; Draxler, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. **Geoscientific model development**, Copernicus Publications G ttingen, Germany, v. 7, n. 3, p. 1247–1250, 2014.
- Christoffersen, P.; Jacobs, K. The importance of the loss function in option valuation. **Journal of Financial Economics**, Elsevier, v. 72, n. 2, p. 291–318, 2004.
- Clevert, D.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In: **4th International Conference on Learning Representations (ICLR’16)**. 2016.
- Delibalta, I. et al. Online anomaly detection with nested trees. **IEEE Signal Processing Letters**, IEEE, v. 23, n. 12, p. 1867–1871, 2016.
- Dey, S. K.; Hossain, A.; Rahman, M. M. Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. In: IEEE. **2018 21st**

international conference of computer and information technology (ICIT). 2018. p. 1–5.

Diouf, N. et al. Channel quality prediction in 5g lte small cell mobile network using deep learning. In: IEEE. **2022 9th International Conference on Soft Computing & Machine Intelligence (ISCFMI)**. 2022. p. 15–20.

Dong, H.; Yang, L. Training robust support vector regression machines for more general noise. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 39, n. 3, p. 2881–2892, 2020.

Fearn, T. Do my data need to be normally distributed? **NIR news**, SAGE Publications Sage UK: London, England, v. 23, n. 1, p. 20–21, 2012.

Forecast, G. et al. Cisco visual networking index: global mobile data traffic forecast update, 2017–2022. **Update**, v. 2017, p. 2022, 2019.

Fu, Y. et al. Artificial intelligence to manage network traffic of 5g wireless networks. **IEEE network**, IEEE, v. 32, n. 6, p. 58–64, 2018.

Gokcesu, K.; Gokcesu, H. Generalized huber loss for robust learning and its efficient minimization for a robust statistics. **arXiv preprint arXiv:2108.12627**, 2021.

Gokcesu, K. et al. Sequential outlier detection based on incremental decision trees. **IEEE Transactions on Signal Processing**, IEEE, v. 67, n. 4, p. 993–1005, 2018.

Gupta, A.; Jha, R. K. A survey of 5g network: Architecture and emerging technologies. **IEEE access**, IEEE, v. 3, p. 1206–1232, 2015.

Hastie, T.; Tibshirani, R.; Wainwright, M. **Statistical learning with sparsity: the lasso and generalizations**. : CRC press, 2015.

He, T.; Droppo, J. Exploiting lstm structure in deep neural networks for speech recognition. In: IEEE. **2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. 2016. p. 5445–5449.

Hodson, T. O. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. **Geoscientific Model Development Discussions**, Göttingen, Germany, v. 2022, p. 1–10, 2022.

Huber, P. J. **Robust statistics**. : John Wiley & Sons, 2004. v. 523.

Intrig-unicamp. **Hackathon Data Collection on 5G Network Usage in Brazil**. 2023. Available at GitHub: .

Jadon, A.; Patil, A.; Jadon, S. A comprehensive survey of regression-based loss functions for time series forecasting. In: Springer. **International Conference on Data Management, Analytics & Innovation**. 2024. p. 117–147.

Khan, F.; Wang, H.; Yang, M. Application of loss functions in process economic risk assessment. **Chemical Engineering Research and Design**, Elsevier, v. 111, p. 371–386, 2016.

Kimura, B. Y. L.; Almeida, J. et al. Deep learning in beyond 5g networks with image-based time-series representation. **arXiv preprint arXiv:2104.08584**, 2021.

- Kurose, J. F.; Ross, K. W. **Computer Networking: A Top-Down Approach**. 7. ed. S.l.: Pearson, 2016. Citado 6 vezes nas páginas 13, 14, 15, 16, 19 e 20. ISBN 978-0-13-359414-0.
- Li, G. et al. Recurrent neural networks based photovoltaic power forecasting approach. **Energies**, MDPI, v. 12, n. 13, p. 2538, 2019.
- Mallinar, N.; Rosset, C. Deep canonically correlated lstms. **arXiv preprint arXiv:1801.05407**, 2018.
- Mansour, A.; Nada, A. E.-R.; Mehana, A. H. Effect of noise variance estimation on channel quality indicator in lte systems. In: IEEE. **2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. 2015. p. 156–160.
- Mathew, A.; Amudha, P.; Sivakumari, S. Deep learning techniques: an overview. **Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020**, Springer, p. 599–608, 2021.
- Medsker, L. R.; Jain, L. Recurrent neural networks. **Design and Applications**, v. 5, n. 64-67, p. 2, 2001.
- Meyer, G. P. An alternative probabilistic interpretation of the huber loss. In: **Proceedings of the ieee/cvf conference on computer vision and pattern recognition**. 2021. p. 5261–5269.
- Mustafa, R. **Benchmarking YouTube in 4G vs 5G Through Open Datasets**. 2023. Available at GitHub: .
- Myttenaere, A. D. et al. Using the mean absolute percentage error for regression models. In: **ESANN**. 2015.
- _____. Mean absolute percentage error for regression models. **Neurocomputing**, Elsevier, v. 192, p. 38–48, 2016.
- Palangi, H. et al. **Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval**. arxiv. org. : February, 2015.
- Parera, C. et al. Transfer learning for channel quality prediction. In: IEEE. **2019 IEEE International Symposium on Measurements & Networking (M&N)**. 2019. p. 1–6.
- Pascanu, R. et al. How to construct deep recurrent neural networks. **arXiv preprint arXiv:1312.6026**, 2013.
- Pham, Q.-V. et al. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. **IEEE access**, IEEE, v. 8, p. 116974–117017, 2020.
- Portnoy, S.; He, X. A robust journey in the new millennium. **Journal of the American Statistical Association**, Taylor & Francis, v. 95, n. 452, p. 1331–1335, 2000.
- Qi, J. et al. On mean absolute error for deep neural network based vector-to-vector regression. **IEEE Signal Processing Letters**, IEEE, v. 27, p. 1485–1489, 2020.

- Qu, Z. et al. Syllable-based acoustic modeling with ctc-smbr-lstm. In: IEEE. **2017 IEEE automatic speech recognition and understanding workshop (ASRU)**. 2017. p. 173–177.
- Raca, D. et al. Beyond throughput, the next generation: a 5g dataset with channel and context metrics. In: **Proceedings of the 11th ACM multimedia systems conference**. 2020, Available: <https://doi.org/10.1145/3339825.3394938>. p. 303–308.
- _____. On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges. **IEEE Communications Magazine**, IEEE, v. 58, n. 3, p. 11–17, 2020.
- Rinne, M. et al. A performance summary of the evolved 3g (e-utra) for voice over internet and best effort traffic. **IEEE Transactions on Vehicular Technology**, IEEE, v. 58, n. 7, p. 3661–3673, 2009.
- Roiger, R. J. Data mining: a tutorial-based primer. CRC press, 2017.
- Sakib, S. et al. A deep learning method for predictive channel assignment in beyond 5g networks. **IEEE Network**, IEEE, v. 35, n. 1, p. 266–272, 2020.
- Sevgican, S. et al. Intelligent network data analytics function in 5g cellular networks using machine learning. **Journal of Communications and Networks**, KICS, v. 22, n. 3, p. 269–280, 2020.
- Silva, L. F. A. e. **Uso de Aprendizado Profundo em Redes 5G através de Representações Pictóricas de Séries Temporais**. Dissertação (Master's thesis) — Universidade Federal de São Paulo – UNIFESP, feb 2022.
- Sun, Q.; Zhou, W.-X.; Fan, J. Adaptive huber regression. **Journal of the American Statistical Association**, Taylor & Francis, v. 115, n. 529, p. 254–265, 2020.
- Vankayala, S. K.; Shenoy, K. G. A neural network for estimating cqi in 5g communication systems. In: IEEE. **2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)**. 2020. p. 1–5.
- Yin, H. et al. Predicting channel quality indicators for 5g downlink scheduling in a deep learning approach. **arXiv preprint arXiv:2008.01000**, 2020.
- Yu, Y. et al. A review of recurrent neural networks: Lstm cells and network architectures. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 31, n. 7, p. 1235–1270, 2019.