TÉCNICO LISBOA

Network Energy Saving Techniques Aided by AI/ML in 4G/5G Networks

André Hilário Cunha

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Luis Manuel de Jesus Sousa Correia

Co-Supervisor: Prof. António Manuel Raminhos Cordeiro Grilo

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino

Supervisor: Prof. Luis Manuel de Jesus Sousa Correia

Members of Committee: Prof. Fernando Corte-Real Mira da Silva

Eng. Diogo Filipe Sequeira Martins

November 2024

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the *Universidade de Lisboa*.

To my friends and family

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to my advisers, Professor Luis Correia and Professor António Grilo, and Engineer Diogo Martins, for their unwavering support and guidance throughout the entirety of this project. I owe special thanks to Professor Luis, who always helped me find the path of least resistance. His insights and advice, paired with his vast knowledge, understanding and consideration, guided me from the beginning to the very end. I also wish to thank Professor António Grilo, whose forward-thinking approach helped me refine and develop the best versions of my ideas. His knowledge and commitment to helping me succeed were invaluable. Lastly, I want to acknowledge Engineer Diogo Martins, who was always available to answer any of my questions, leaving no doubts unanswered. His dedication and willingness to go the extra mile in every process was truly appreciated. Together, their consistent presence and commitment, meeting with me every week without fail, ensured that I could achieve the best possible outcome for this thesis, and I could not have asked for any better.

I would also like to thank my family, whose support has been constant and invaluable. To my mom, thank you for taking responsibilities off my shoulders, allowing me to fully focus on my work. Even when the topics were too complex to follow, you always listened, encouraged, and reassured me with your pride. To my dad, thank you for making sure everything was in order, from emotional support to building my confidence, ensuring I had everything I needed to succeed and always being a role model to follow. To my sister, thank you for being my sounding board in our daily talks, always offering a space to share my worries and helping me feel better afterward. To my little brothers, thank you for bringing out the best in me, whether it was through playing football, video games or just about anything we did together. You reminded me of the joys outside of my work, and I'm grateful for your constant companionship and admiration. To my grandmothers, thank you for all the dinners you cooked for me, for all the times you endured my absence and reassured me of your love. To my grandfather, my biggest friend and inspiration, no words are needed, nor can they explain the feelings we share. You remain to this day the foundation upon which I have built myself as a person, and in a sense, this was only possible through you.

To my close friends – Bernardo, Luis, Rosa, Raquel, David, Raquel, Leonor, Pedro, Gonçalo, Hugo, Daniel, and Simão – thank you for always sticking by me, especially during the times when I could not join you because of deadlines or meetings. Your concern for my well-being and encouragement kept me going through the most stressful moments with a big happy smile.

Finally, to my girlfriend and best friend, Raquel, thank you for being there every step of the way. You shared both the joys and challenges of this journey with me, making even the hardest times bearable. Thank you again, for always ensuring I was ok, even when I did not want to admit it, for guiding me though all the hard choices life threw at me and for being the voice of reason I very much need. Your love, support, and belief in me kept me grounded and motivated, and for that, I will always be grateful.

Abstract

This thesis aims to study the aspects and characteristics of 4G and 5G mobile networks, their energy consumption and potential optimization using energy efficiency techniques with the aid of machine learning. A traffic dataset was provided by Vodafone, which was carefully analyzed to identify potential energy-saving opportunities that could be simulated. A time-based energy efficiency technique is developed aimed at turning off components of the base station during periods of low or no traffic. To predict these periods, an algorithm was developed, leveraging traffic predictions from an LSTM model to guide energy-saving actions. The algorithm is designed to deliver substantial energy efficiency while also being customizable to meet the specific requirements of network operators and remaining easy to implement and maintain. The LSTM model effectively predicts future sector usage by accurately learning and adapting to daily network usage patterns, significantly enhancing the effectiveness of energy-saving actions performed by the algorithm. The proposed algorithm is able to achieve high predictive accuracy, whose outcomes demonstrate substantial results, achieving up to 2,451,624 kWh of saved energy, translating to total monetary savings of €403,047 and total CO2 emissions saved of 405.7 tons in a year of simulated use across the whole 5G network of the country in study.

Keywords

5G, 4G, Energy Efficiency, Machine Learning, LSTM, Traffic Prediction, Network Optimization.

Resumo

Esta tese tem como objetivo estudar os aspetos e características das redes móveis 4G e 5G, o seu consumo de energia e a potencial otimização através de técnicas de eficiência energética com o auxílio de machine learning. Foi fornecido um conjunto de dados de tráfego pela Vodafone, que foi cuidadosamente analisado para identificar possíveis oportunidades de poupança de energia que poderiam ser simuladas. Uma técnica de eficiência energética baseada no tempo foi desenvolvida, destinada a desligar componentes da estação base durante períodos de tráfego baixo ou nulo. Para prever esses períodos, foi desenvolvido um algoritmo que utiliza previsões de tráfego de um modelo LSTM para orientar ações de poupança de energia. O algoritmo foi desenhado para oferecer uma eficiência energética substancial, sendo também personalizável para atender aos requisitos específicos dos operadores de rede, além de ser de fácil implementação e manutenção. O modelo LSTM prevê com eficácia o uso futuro dos setores ao aprender e adaptar-se com precisão aos padrões diários de utilização da rede, aumentando significativamente a eficácia das ações de poupança de energia realizadas pelo algoritmo. O algoritmo proposto alcança uma elevada precisão de previsão, com resultados que demonstram um impacto substancial, atingindo até 2.451.624 kWh de energia poupada, o que se traduz em poupanças monetárias de €403.047 e numa redução de emissões de CO2 de 405,7 toneladas por ano de uso simulado em toda a rede 5G do país em estudo.

Palavras-chave

5G, 4G, Eficiência Energética, Aprendizagem Automática, LSTM, Previsão de Tráfego, Otimização da Rede.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
5GC	5G Core
AAS	Active Antenna System
AAU	Active Antenna Units
AF	Application Function
AMF	Access and Mobility Management Function
AN	Access Network
ANN	Artificial Neural Network
ASM	Autonomous Sleep-Modes
AUSF	Authentication Server Function
BBU	Base Band Unit
BFE	Backward Feature Elimination
BTS	Base Transceiver Station
BWP	Bandwidth Part
CP	Cyclic Prefix
CriC	Critical Communications
DFT-s-OFDM	Discrete Fourier Transform Spread-OFDM
DL	Downlink
DNN	Deep Neural Network
DQN	Deep Q-Network
DRB	Data Radio Bearer
DRX	Discontinuous Reception
DTX	Discontinuous Transmission
ECR	Energy Consumption Rating
ECR-EX	ECR Extended Idle Load Cycle
ECR-VL	ECR Variable Load
ECRW	ECR-weighted
eMBB	Enhanced Mobile Broadband
eNB	Evolved NodeB
en-gNB	Next-Generation NodeB
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
eVX2	Enhanced Vehicle to Everything
FDD	Frequency Division Duplex
FFS	Forward Feature Selection
GA	Genetic Algorithm
GHG	Greenhouse Gas
LSTM	Long Short-Term Memory
MDA	Mean Decrease Accuracy
MDI	Mean Decrease Impurity
MDP	Markov Decision Process
MIMO	Multiple Input, Multiple Output

MIoT	Massive Internet of Things
ML	Machine Learning
MLP	Multi-Layer Perceptron
MME	Mobility Management Entity
mMTC	Massive Machine Type Communications
MN	Master Node
MU-MIMO	Multi-User MIMO
NEF	Network Exposure Function
NG-RAN	Next-Generation Radio Access Network
NRF	Network Repository Function
NSA	Non-Standalone
NSSF	Network Slice Selection Function
OFDMA	Orthogonal Frequency Division Multiple Access
OPEX	Operating Expenses
PA	Power Amplifier
PAPR	Peak-to-Average Power Ratio
PCF	Policy Control Function
PDF	Probability Density Function
PDN-GW	Packet Data Network Gateway
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RE	Resource Element
RF	Radio Frequency
RL	Reinforced Learning
RNN	Recurrent Neural Network
RRC	Radio Resource Control
RRU	Remote Radio Units
SA	Standalone
SC-FDMA	Single-Carrier Frequency Division Multiple Access
SCP	Service Communication Proxy
SCS	Subcarrier Spacing
SFS	Sequential Feature Selection
S-GW	Serving Gateway
SINR	Signal-To-Interference-Plus-Noise Ratio
SMF	Session Management Function
SN	Serving Node
SNR	Signal-to-Noise Ratio
SRB	Signaling Radio Bearer
SUL	Supplementary Uplink
SU-MIMO	Single-User MIMO
TDD	Time Division Duplex
TEEER	Telecommunications Equipment Energy Efficiency Rating
TEER	Telecommunications Energy Efficiency Ratio
TRX	Transceiver
TTI	Transmission Time Interval

UDM	Unified Data Management
UE	User Equipment
UL	Uplink
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communications
XGBoost	Extreme Gradient Boosting

List of Symbols

A	Accuracy
$d(x, y)$	Euclidean Distance
I	Inertia
$I(X;Y)$	Mutual Information
I_{IN}	Inertia
I_{CI}	Confidence Interval
Max_{RCC_SA}	N_RCC_Users_SA_Max
$Max^{DL}_{activeUE}$	Max_N_UE_DL
$Max^{UL}_{activeUE}$	Max_N_UE_UL
$N_{PRB_available}$	N_PRB_available
N_{RBsym_A}	N_RBsym_typeA
N_{RBsym_B}	N_RBsym_Broadcast
N_{RBsym_CSI}	N_RBsym_Signaling
N_{RBsym_Total}	N_RBsym_available
$N^{DL}_{activeUE}$	N_UE_DL
$N^{DL}_{activeUE}$	N_UE_UL
Psine	Sine Period
$R1-4_{CQI}$	CQI_Rank_1-4
$\mathrm{sigmoid(x)}$	Sigmoid Function
$SINR_{PUCCH}$	SINR_PUCCH
$SINR_{PUSCH}$	SINR_PUSCH
$\tanh(\mathrm{x})$	Hyperbolic Tangent Function
Thp_{DL}	THP_User_NonGBR[Mbps]
T_{DL_active}	T_DL_active
T_{DL}	T_DL
$T_{DL\_active\_s}$	T_DL_Active_s [s]
$U_{ReLu}(x)$	ReLu Function
V_{DL}	Vol_DL_Data
$V_{DL\_Data\&Signaling}$	Vol_DL_Data&Signaling
V_{DL_MAC}	Vol_DL_MAC [MB]
ρ	Pearson's Correlation
ρ_{a}	Autocorrelation
ρ_{s}	Spearman's Correlation

List of Software

Python (version 3.10.11)
MS Word (version 2409)
MS Excel (version 2409)
Chat-GPT-4o
Kaggle
Draw.io

Chapter 1

Introduction

This chapter provides a brief overview of the topic, focusing on the existing problems that justify the work, objectives, and motivations of the work, along with a small description of the contents of the thesis.

1.1 Overview and Motivation

In recent years, wireless networks have gone through great advancements, becoming an indispensable part of our everyday lives. This rapid evolution has brought about its widespread adoption, from giving connection to the internet to enabling a vast range of services and applications, making it one of the foundations of our connected world. This rapid evolution and adoption also lead to a rapid increase in subscribers and, in turn, cellular traffic. Mobile subscribers are expected to grow from 5.1 billion in 2018 to around 6.7 billion by the end of 2023 [TaPe19] which translates to 15 billion connected devices worldwide and 130 EB per month of mobile data traffic by the end of 2023, which totals 1.5 ZB in 2023, with forecasts showing this increasing trend continuing in the near future, as shown in Figure 1.1 [VaLS23], [Eric23].

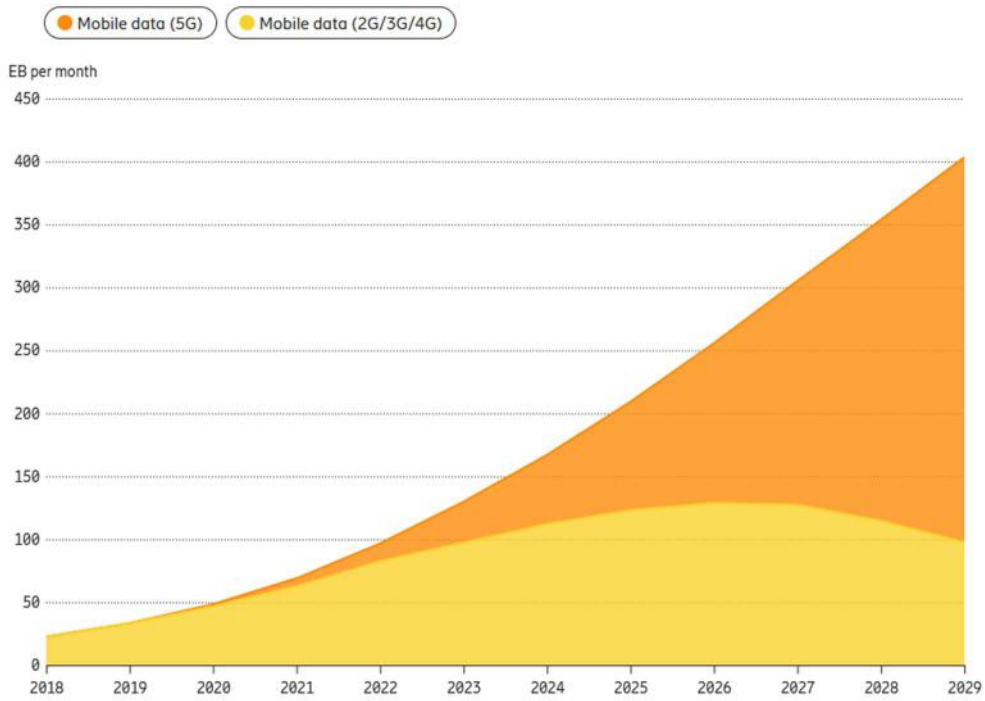


Figure 1.1 - Global mobile network data traffic (extracted from [Eric23]).

This rapid expansion leads to an even faster increase in mobile network energy consumption and overall operational expenses for network providers, as well as the increase of greenhouse gas emissions since most of the energy used comes from fossil fuels.

The use of fossil fuels in different industries is causing an increasing rise in greenhouse gas emissions which consequently causes global warming. As the global climate is in a state of emergency, any measures taken to turn it around are of the most importance, such as this work. Efficiencies in mobile networks do not only reflect in emissions contribution of its industry. Since almost all industries utilize mobile networks, and with the rise of service support with 5G more are going to be able to adopt a wider use, all efficiencies in mobile networks will translate to even greater ones in other industries, with the introduction of concepts from unmanned operations, resulting in greater work efficiency, to remote work resulting in less transportation pollution.

As the demand for fast and seamless wireless connectivity increases, so does the need to address the problems associated with it. Multiple initiatives and targets have been proposed by international bodies to reduce GHG (Greenhouse Gas) emissions and address climate change. Among them, the United Nations Framework Convention on Climate Change - the Paris agreement, the EU European Green Deal, the C40 Cities Global Green New Deal and IMO Marine Environment Protection Committee all share similar goals and targets: 40-50% reduction of GHG emissions by 2030-2050 [CHAC20].

Currently, around 25% of the operating expenses (OPEX) of the network providers is spent on the networks, which will keep increasing as 5G starts getting adopted, and 90% of it is spent in energy [THEK20]. In a mobile network, up to 70% of energy is used by the Base Station (BS) [KHSM21], making it the component where largest efficiencies can be achieved, which will be the focus of this thesis. Figure 1.2 shows a predicted breakdown of energy consumption by network level in 2025.

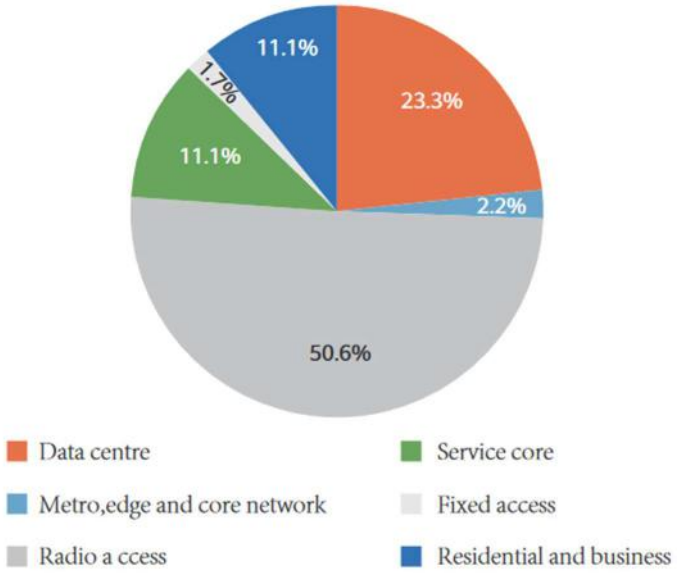


Figure 1.2 - Energy consumption breakdown by network element in 2025 (extracted from [CHAC20]).

With this goal in mind, along with the incentive of reducing operational costs of providers, the research and development of techniques aimed at reducing overall network energy consumption increased. The development and design of 4G and 5G networks took all this into account: enabling this forecasted growth and, especially in 5G, enabling more and better network energy efficiency techniques to be developed and implemented.

To aid such techniques, emerging technologies such as machine learning have garnered attention. The ability of Machine Learning to analyze vast datasets, predict network traffic patterns, and dynamically adapt to changing conditions makes it a valuable tool in achieving energy savings in the network. For instance, ML algorithms can anticipate network traffic fluctuations, enabling pre-emptive measures to be taken to optimize power consumption. These algorithms can also adaptively scale resources, supporting a steady influx of new functionality while dynamically adjusting to varying levels of demand.

The focus of this thesis is on developing energy-saving techniques for base stations using machine learning, specifically LSTM models. By leveraging traffic predictions, preemptive measures are

implemented during periods of low or no traffic, aiming to reduce energy consumption with minimal impact on network performance.

1.2 Thesis Structure and Contents

The objective of this thesis is to develop a technique aimed at reducing energy consumption at the base station while leveraging machine learning, with the goal of contributing to environmental sustainability and alleviating financial burdens on network operators. To accomplish this, a technique will be developed following these steps:

- Data exploration and prepossessing: The data is thoroughly examined to gain familiarity with its structure and the potential opportunities for implementing energy efficiency techniques. This process involves exploring various aspects of the dataset, such as traffic patterns, network conditions and quality of service metrics, to identify areas where energy-saving measures could be applied. Additionally, the data is carefully organized and preprocessed to ensure it is properly prepared for integration into the algorithm.
- Technique selection: Identify the most suitable energy-saving actions that the algorithm can implement, given the available data.
- Model selection and training: Determine the optimal LSTM model architecture that ensures accurate and robust traffic predictions, enabling more effective energy management through well-timed and precise energy-saving actions.
- Result analysis: Conduct a detailed analysis of the results obtained from the LSTM model, evaluating its prediction accuracy, energy-saving effectiveness, and overall performance and comparison to baseline models and alternative approaches.

The thesis is composed of 5 chapters. Chapter 1 introduces the main topic of this thesis, making a brief overview of the topic along with the problems and motivations for the development of a solution. Chapter 2 brings about the fundamental concepts of this work, the 4G/5G systems, their energy consumption and efficiency aspects and an introduction to the different types of machine learning and algorithms that will be used in the development of the thesis. Chapter 3 focuses on the development and proposal of the energy-saving algorithm, detailing all methods used, its functioning, and the thought process behind its design. Chapter 4 analyzes the results of the algorithm, exploring varied scenarios and showcasing the possible optimizations and customizations available, and, finally, Chapter 5 provides a concise overview of the thesis, summarizing the key findings and insights gained throughout the research and developed process. Additionally, this chapter presents the conclusions drawn from the study, highlighting the contributions made to the understanding of energy efficiency and the potential for further research and development in this area.

Chapter 2

Fundamental Concepts

This chapter provides an overview of the 4G/5G systems, mainly focusing on their architecture, radio interface, energy consumption and efficiency. It also introduces basic concepts of machine learning and presents a state-of-the-art analysis on the main topic.

2.1 The 4G/5G systems

This section goes over the main aspects of the 4G/5G systems relevant for this thesis. Section 2.1.1 covers the network architecture of these systems and Section 2.1.2 the radio interfaces.

2.1.1 Network Architecture

This section is based on [3GPP19], [3GPP23]. There are two different deployment architectures for 5G: Non-Stand Alone (NSA) and Stand Alone (SA). The NSA architecture uses the 5G Access Network (AN) and its New Radio (NR) interface alongside the existing 4G core and radio, effectively being 4G with the extra radio capabilities of 5G, while the SA architecture only uses the 5G components, where the 5G AN is directly connected to the 5G core (5GC).

The NSA architecture is represented in Figure 2.1.

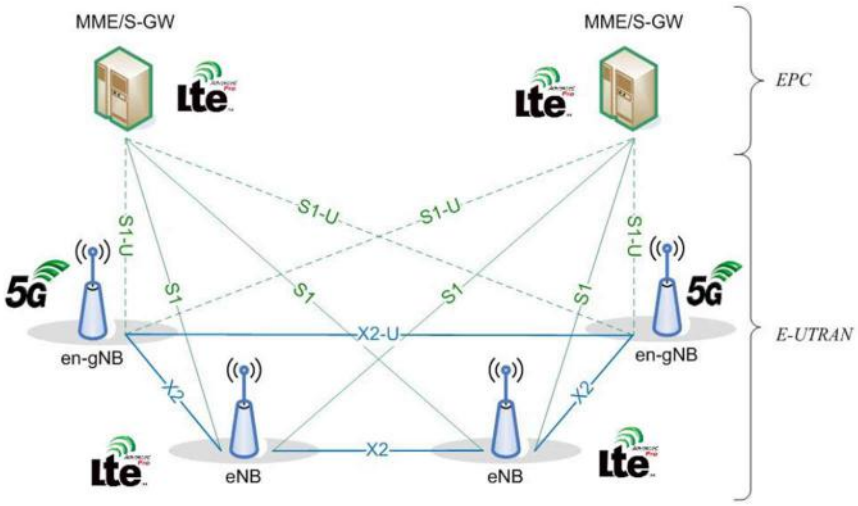


Figure 2.1 - NSA network architecture (extracted from [3GPP19]).

The NSA architecture serves as a middle step between 4G and “full” 5G. As it can be built upon an existing 4G infrastructure, it makes 5G available without the need to replace the network. For the same reason, NSA only supports 4G services as it uses a 4G core, while still getting the NR benefits for using 5G base stations (en-gNB). The 5G base stations are connected to the 4G base stations (eNB) through the X2 interface.

As seen in Figure 2.1, the NSA architecture is composed of the Evolved Packet Core (EPC), and the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), which is the AN of 4G and is composed of the different base stations and connection interfaces. The EPC is the core network component in LTE. It includes elements like the Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Data Network Gateway (PDN-GW) that manage functions such as mobility management, packet routing, and connectivity to external networks. The EPC is connected to the base stations in the E-UTRAN using the S1 and S1-U interface, for eNB and en-gNB, respectively. The NSA architecture provides dual connectivity to both 4G and 5G AN. This way it can also be called "EN-DC",

for "E-UTRAN and NR Dual Connectivity". In this configuration, the 4G eNB is the Master Node (MN) while the 5G en-gNB is the Secondary Node (SN).

The SA architecture is shown in Figure 2.2.

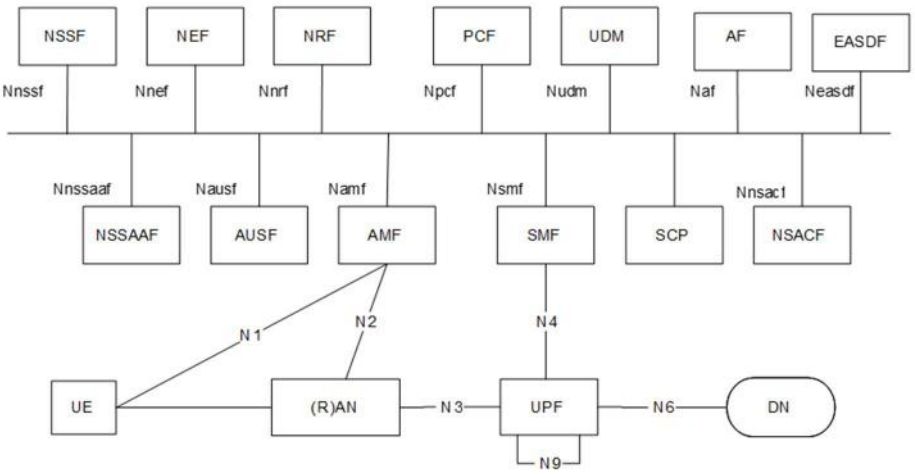


Figure 2.2 - SA network architecture (extracted from [3GPP23]).

In the SA architecture, the 5G AN, called New Generation Radio Access Network (NG-RAN), connects the user equipment (UE) directly to the 5GC. SA 5G has a service-based architecture, where the 5GC manages separate Control Plane and User Plane functions. This separation enables them to evolve independently, making it easier to scale, upgrade, and deploy network functions as needed. The User Plane Function (UPF) is a crucial element of a 5G network, responsible for managing user data, enforcing policies, ensuring quality of service, and serving as a gateway for external networks. The Control Plane functions are as follows:

- Access and Mobility Management Function (AMF): The AMF is responsible for managing the connections between the core and the NG-RAN and the UE, registration, connection and mobility and support for Network Slice-Specific Authentication and Authorization.
- Session Management function (SMF): The SMF is responsible for session management and establishment, by allocating and managing IP addresses of the UE, as well as managing UPF sessions.
- Policy Control Function (PCF): The PCF supports a unified policy framework to govern the network behavior and provides policy rules to Control Plane functions.
- Network Exposure Function (NEF): The NEF serves to expose network capabilities and events, communicate with external applications in a secure manner and translate information between external applications and internal network functions.
- Network Repository Function (NRF): The NRF keeps an up-to-date repository of the network functions and their capabilities, as well as providing that information to help the different network functions to discover and communicate with the appropriate services and functions.
- Unified Data Management (UDM): the UDM is responsible for managing user data, profiles, authentication, and subscriber-related information to ensure the security and quality of service delivered to the user.

- Authentication Server Function (AUSF): The AUSF is responsible for verifying user and ensuring that only authorized users and devices access the network.
- Application Function (AF): The AF is responsible for interacting with the Core Network and managing a wide range of services.
- Network Slice Selection Function (NSSF): The NSSF is responsible for selecting the set of Network Slice instances serving the UE and determining the AMF that should serve the UE.
- Service Communication Proxy (SCP): The SCP acts as an intermediary for communication between network functions and external services, providing translation, security, routing, and other functions.

2.1.2 Radio Interface

This section is based on [3GPP17] and [Laun21]. Both 4G and 5G use multiple access techniques that allow multiple users to share the same communication channel or radio spectrum simultaneously. Both use Orthogonal Frequency Division Multiple Access (OFDMA) for the downlink (DL), which, in contrast with 4G, can also be used as uplink (UL) in 5G along with OFDM with Discrete Fourier Transform precoding (DFT-s-OFDM), the latter being used to reduce peak-to-average power ratio (PAPR) to achieve higher UL coverage with lower energy consumption of the UE, with the downside of being less spectrally efficient. 4G uses Single-Carrier Frequency Division Multiple Access (SC-FDMA) for UL, for the same reason mentioned before.

Two frequency ranges are defined for 5G: FR1 and FR2. The FR1 frequency range covers the bands between 450 MHz and 7.125 GHz, with channel bandwidth between 5 and 100 MHz. FR2 covers the bands between 22.45 and 52.6 GHz, with channel bandwidth between 50 and 400 MHz. 4G operates in between the 450 MHz and 3.8 GHz frequency ranges with a channel bandwidth starting from 1.4 MHz up to 20 MHz.

Subcarrier spacing (SCS), Δf, is a critical parameter in OFDMA that determines the efficiency and performance of the system. For 4G, the SCS is fixed at 15 kHz, with a maximum of 1200 subcarriers, while in 5G this value can change depending on the parameter μ, using (2.1):

$$\Delta f = 2^{\mu} \times 15\ [\mathrm{kHz}]\ , \tag{2.1}$$

with a maximum of 3 300 subcarriers. The parameter μ, which can also be referred to as numerology, can assume values from 0 to 4: 0 or 1 for FR1, 3 or 4 for FR2 and 2 for both.

Duplexing is used by both 4G and 5G. Duplexing is used to enable both UL and DL communications on the same channel. There are two main duplexing techniques used by both networks: Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD). FDD uses two different frequency bands for UL and DL transmission, separated by a guard band. These values are determined by the base station and sent to the UE. TDD sends both UL and DL transmissions in the same frequency band but at different time slots. Additionally, 5G offers Supplementary Uplink (SUL), where a new frequency band is used for UL, generally using a low frequency to increase coverage and latency for the UL. 4G uses both TDD and FDD for its frequency range, while 5G uses FDD, TDD and SUL on FR1 and only TDD on FR2.

4G and 5G transmissions are arranged into frames, each divided into 10 subframes. as shown in Figure 2.3.

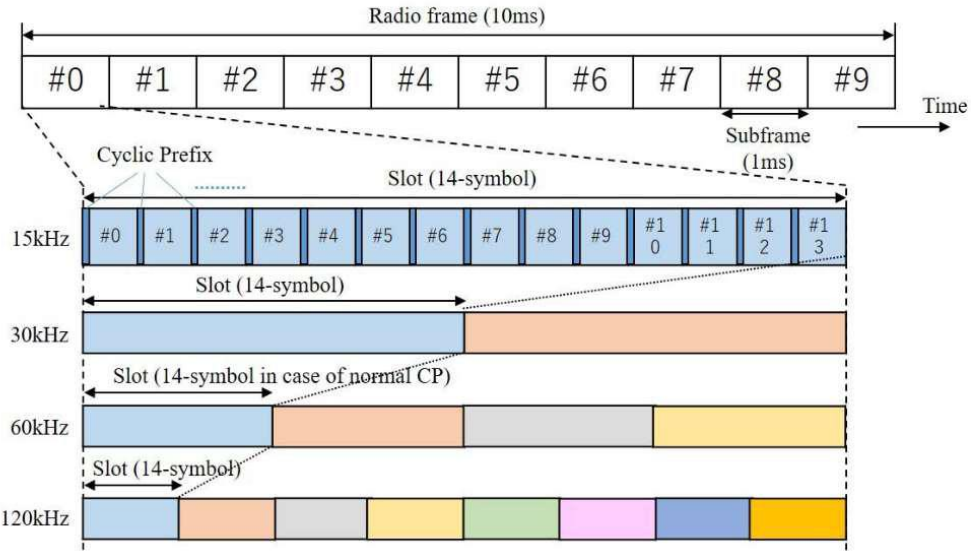


Figure 2.3 - 4G and 5G radio frame representation (extracted from [3GPP19]).

Each subframe has 1 ms duration, resulting in 10 ms frames. In 4G, each subframe consists of 2 slots, each with a 0.5 ms duration. In 5G, the number of slots per subframe and their duration, Transmission Time Interval (TTI), depends on the SCS. These values are presented in Table 2.1. This allows 5G to be able adapt depending on the desired latency.

Table 2.1 - Structure of the 5G frame in the time domain (extracted from [Laun21]).

Subcarrier Spacing (kHz)	Number of slots per subframe	Number of slots per frame	TTI (ms)
15	1	10	1
30	2	20	0.5
60	4	40	0.25
120	8	80	0.125
240	16	160	0.0625

In 5G, each slot consists of 14 OFDM symbols preceded by a cyclic prefix (CP). For 60 kHz, there is a possibility of having an extended CP, in which case the slots consist only of 12 symbols. The extended cp is four times longer than the normal one and is used for cells that have a large delay spread. For 4G, each slot contains 7 OFDM symbols with normal CP and 6 with an extended one.

Both networks use resource elements (RE) and resource blocks (RB) to allocate time and frequency resources for data transmission, ensuring optimal utilization of the radio spectrum while accommodating multiple users and different communication needs. A RE is a two-dimensional unit composed of one OFDM symbol in the time domain and one subcarrier on the frequency domain. An RB is a group of RE,

in this case composed of one slot and 12 subcarriers. The number of available RBs changes with the bandwidth and, in case of 5G, also with numerology.

Multiple input multiple output (MIMO) is a technique where multiple transmitting and receiving antennas are used together to form an array, allowing for multi-layer data transmission, leading to higher throughput, network reliability and better coverage. Both 4G and 5G support this technology for a single user (SU-MIMO), where multiple antennas transmit to a single user, and for multiple users (MU-MIMO), where the same happens for more than one user. MIMO also enables the use of beamforming, where the antennas can control the direction of the beam by changing their amplitude and phase accordingly. This reduces the beam opening angle, leading to a stronger signal with better coverage, lower interference, and better energy efficiency, since unnecessary radiation directions are removed. To achieve this, active antenna systems (AAS) are used to allow for both horizontal and vertical beamforming, allowing for better focus on the users, as seen in Figure 2.4. To allow for narrower and better focused beams and service of more users, massive MIMO were developed. Massive MIMOs are scaled up versions of MIMO. While MIMO uses up to 16 antennas, massive MIMO can have hundreds of them, greatly improving signal quality, coverage, and capacity.

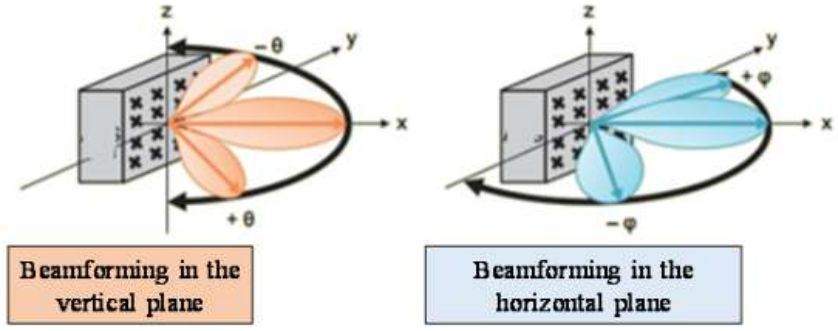


Figure 2.4 - Beamforming in horizontal and vertical planes using MIMO (extracted from [Laun21]).

2.2 Energy Aspects in 4G/5G Networks

This section breaks down aspects of 4G/5G systems power consumption and efficiency in Section 2.2.1, as well as reviewing some energy efficiency techniques in Section 2.2.2 and metrics in Section 2.2.3.

2.2.1 Power Consumption and Network Energy Efficiency

The 4G and 5G mobile networks need energy to power all its components and ensure good performance. In a mobile network, the radio access network (RAN), responsible for the wireless transmission of data, is what consumes the most power, more than half and up to more than 70% of total power consumed [KHSM21], [CHAC20]. Therefore, from all the components present in the network, the base station accounts for most of the consumed power.

The BS is comprised of multiple transceivers (TRX), each serving one transmit antenna. A TRX comprises a power amplifier (PA), a radio frequency (RF) small-signal TRX module, a baseband engine

including a receiver (uplink) and transmitter (downlink) section, a DC-DC power supply and an active cooling system. Core energy consumption in a mobile network refers to the energy used by essential network components and functions, along with all energy consumption associated with backhaul transport. Data center energy consumption refers to the energy used by data centers where the IT systems of an operator and intranet infrastructure are hosted, [AGDG12], [KHSM21].

High power consumption in the RAN makes energy efficiency in the BS a crucial and extensively discussed topic in mobile networks. BS resources are generally unused 75 - 90% of the time. This is primarily because most of the hardware components in the BS remain operational to transmit idle mode signals, as stipulated by the 4G or 5G standards, such as synchronization and reference signals, and system information. Deployed 5G networks are more efficient than 4G ones because of its leaner carrier design, where the amount of unnecessary idle mode signaling are reduced: In 4G a BS must transmit reference signals about 1000 times per second while in 5G there can be transmission-free time slots leading to 5-100 ms intervals without transmission, enabling sleep modes, where some components of the BS are shut off, greatly increasing 5G overall efficiency and enabling for more techniques to be implemented. Another reason is the wider use of MIMOs and massive MIMOs, that as explained in Section 2.1.2, also increases the networks energy efficiency [SRBM21].

A sleep mode is the deactivating of system components when there is no activity in the BS, meaning periods where no data needs to be transmitted or received and there are no synchronization signals to be sent. Depending on the durations of these periods, deeper states of sleep, advanced sleep modes, can be achieved, where more critical components can be switched off gradually, when the hardware present in the BS allows for their shut down and power up in a very short period. This can be seen in Figure 2.5, where a graph of power consumption is shown while different sleep modes are activated [SGAC17], [SRBM21], [PDPX22].

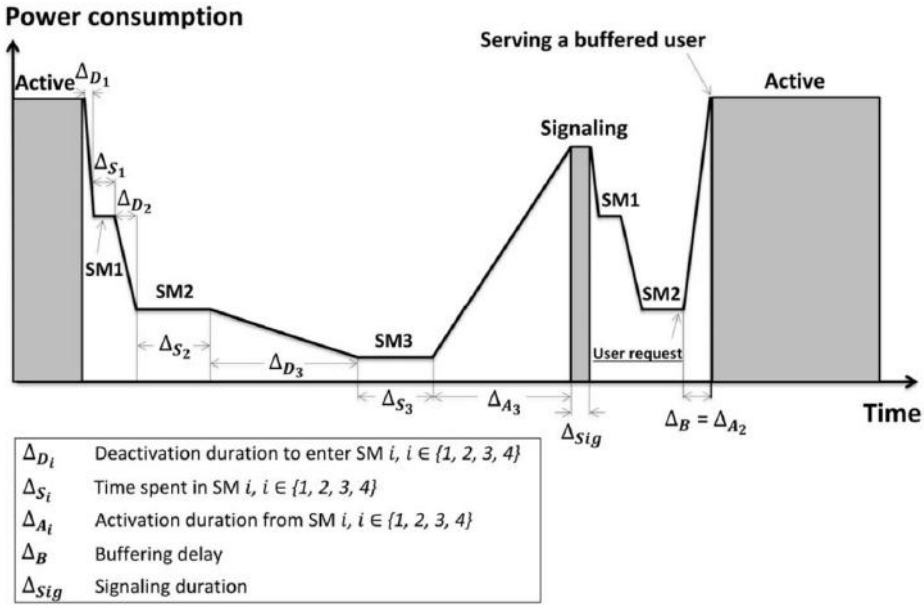


Figure 2.5 - Example of Sleep Mode implementation (extracted from [SGAC17]).

Other energy saving techniques aim to maximize how long and deep a BS sleep is, as well as minimizing the resources utilized while keeping QoS, thus maximizing the amount of time BS components can be off, ultimately saving energy.

Even so, 5G can consume up to 140% more power than 4G, as achieving the same coverage using higher frequency bands leads to a higher number of cells needed, as well as higher power needs to achieve higher bandwidth and signal throughput [KHSM21], [PPDG22], making energy saving techniques even more important to develop.

2.2.2 Energy Efficiency Enabling Techniques

In this section some techniques adopted in 3GPP are discussed, as well as some proposed solutions for 5G networks.

In 4G, 3GPP supports small cells deployment along macro-cells, to function as booster cells. These small cells are placed within a macro-cell coverage range, particularly at cell edge, to improve network capacity, there. When the load in the main cell is low, the small booster cell can be shut off by it to reduce energy consumption. When this occurs, UE connected to the booster cell are handed over to the main cell, while neighboring BS are also notified of the booster cell switching off. The same happens the other way around. A similar procedure, defined in 3GPP, takes place in NSA with EN-DC with the 4G eNB as the primary node and the 5G gNB as the secondary one. In this scenario, the 4G eNB can also switch the 5G gNB on and off as the load varies, utilizing the gNB as the booster cell in this case. UEs connected to this BS are also informed of this, similarly to the first scenario. Additionally, eNB can use cell muting, where the DL of a cell is turned off, while maintaining the transmission of reference signals. While this technique is usually used to reduce interference, it can also be utilized to reduce overall energy consumption [ILSL23]. The use of MIMOs, as stated in Section 2.2.1, can also enhance energy efficiency by improving coverage, signal strength, throughput and many others already covered in Section 2.1.2, as well enabling other techniques that will be covered next.

As previously stated, 5G enables for more techniques to be developed, as there is greater potential to explore BS inactivity because of the ability to lower the frequency of idle mode signals, enabling BS sleep intervals along with dynamic resource usage base on traffic load.

As explored in [ILSL23] potential techniques proposed to enable energy saving in 5G can be categorized into four domains: time, frequency, spatial and power. These techniques aim to reduce BS activity in their respective domain.

Time domain techniques reduce the time of transmission and reception of data to allow for longer BS inactivity. To do this, the BS can enter an energy saving state, where a discontinuous reception/transmission (DRX/DTX) cycle is set up. A DRX cycle is a power saving feature where a device, in this case the BS, is only active to receive and transmit data during a set period in between a period of inactiveness, as shown in Figure 2.6.

This maximizes the time the BS can enter sleep modes, as well as allowing them to be deeper and more energy saving. While this technique works with active UEs, since UEs also have DRX implemented to

save energy, paging still needs to be performed by the BS, potentially waking it up prematurely. Figure 2.7 illustrates legacy paging procedure, where paging frames are distributed over time, greatly reducing the depth and length of BS sleep.

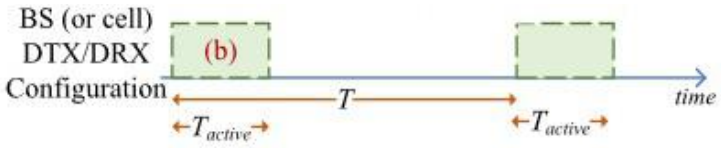


Figure 2.6 - Example of BS DTX/DRX duty cycle (extracted from [ILSL23]).

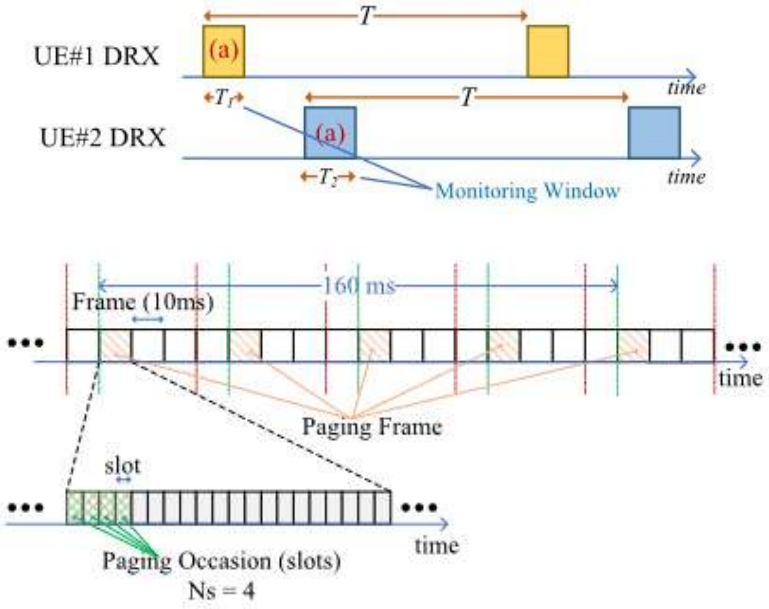


Figure 2.7 - Typical UE DRX cycles and corresponding BS paging occasion allocation (extracted from [ILSL23]).

With the same goal of maximizing the period of inactiveness of the BS, paging frames could be sent out in a more efficient manner, in this case, condensing them into consecutive slots or frames, while maintaining the same paging density. This is illustrated in Figure 2.8.

Frequency domain techniques reduce frequency usage while BS is transmitting, by reducing the operating BW of a carrier, changing the number of active carriers, adapting transmissions between carriers, etc. While using carrier aggregation, there is no need to send synchronization signals in all carriers involved. Another way of saving energy is by using 3GGP support for SSB-less cell operation for intra-band carrier aggregation. This means that when some carriers share the same band, an anchor cell can manage other idle mode signals of non-anchor cells. The same can be extended for intra-band carrier aggregation. With this technique, when load allows it, some cells may be able to enter very deep sleep states. 5G also introduces bandwidth parts (BWP), where the existing carrier bandwidth can be divided into smaller portions dedicated and tailored to specific services or users. This allows for more efficient use of the bandwidth and over lower energy consumption. While reducing available bandwidth may decrease energy use, in certain scenarios, depending on the use of the network, this can lead to

greater times being used for transmission, which can potentially hurt energy efficiency, so a trade-off between BW and load must be accomplished.

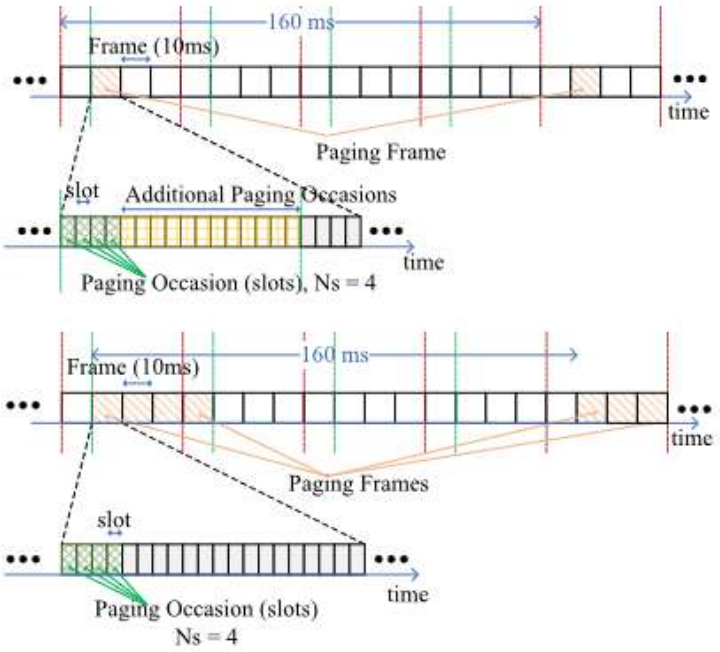


Figure 2.8 - Potential enhancement by compacting paging frames (extracted from [ILSL23]).

Spatial techniques work by turning on/off physical components on the BS, usually antennas and the PA associated with them. One, multiple or even all antennas can be shut off, depending on the load, greatly decreasing energy usage. Depending on the number of antennas that are shut off, some UEs may lose coverage for that cell, which is non ideal as it causes delays in UE receptions, as they are dependent on BS paging to wake up from their idle mode. Another impact caused by this technique is the change in possible beam formation and gain, as well as muting of reference signals if one antenna responsible for it is shut off. Dynamic changes to antenna port configuration for various reference signals is not currently supported in 5G, and all changes must be made using Radio Resource Control (RRC) configuration procedures, which has latencies in the order of 10 ms [3GPP20]. Reducing the number of antennas also potentially reduces both potential diversity and spatial multiplexing gains and user throughput which can, again, increase the duration of transmission, hurting energy efficiency. Therefore, careful trade-offs must be considered with the use of this technique, to ensure QoS. This means that these techniques are more suited to lightly loaded BS, where these changes will not be so noticeable.

Power domain techniques try to reduce DL transmission power and improve PA efficiency, while keeping data transmission as little impacted as possible. These techniques require careful management because changing the power levels can affect signal quality. For example, there is a need to ensure that the network and UE agree on the power offset between measurement signals and data channels. If this is not done accurately, it can lead to performance issues and negate the energy-saving benefits. To address this, 3GPP is working on dynamic power adaptation methods. These methods involve improving the configuration of the UEs and enhancing the feedback from UEs to help save energy more effectively. Additionally, as networks move to higher frequencies, PAs become less efficient, so techniques like

digital pre-distortion and tone reservation are being explored to make power amplification more efficient and reduce signal distortion.

All these techniques can be greatly improved with the use of AI and ML, which is the core objective of this work, as these systems can be a big help in deciding when and how these should and should not be applied.

2.2.3 Energy Efficiency KPI

To better understand and compare the efficiency of a mobile network, metrics and KPIs need to be discussed. Various metrics can be chosen depending on the use cases and the purpose of the analysis and can focus on facility-level, equipment-level, and network-level efficiency [HBVB11]. For this work, only equipment and network level metrics will be discussed as these will be the focus of our study.

Equipment-level metrics evaluate the energy efficiency of a specific equipment, like BSs. These metrics usually measure the ratio between energy consumption with different performance parameters. The energy consumption rating (ECR), [MCIT15], is one widely adopted metric around the world that relates energy consumption with maximum throughput. This metric, however, does not consider that networks and equipment do not always operate at maximum load. For this reason, more metrics were developed from the ECR to encompass the dynamic network conditions, like full, half, and idle loads.

Other metrics are then introduced to cover these points, where the energy consumption calculation is changed to match each network condition like the ECR-weighted (ECRW), energy efficiency metric over a variable-load cycle (ECR-VL) and energy efficiency metric over extended-idle load cycle (ECR-EX) [HBVB11]. Other useful metrics are EER, which is defined as being the inverse of ECR, power per user, telecommunications energy efficiency ratio (TEER), which is the ratio of useful work to power consumption, where useful work refers to beneficial output, like transmitting data, supporting calls or handling network traffic efficiently, and telecommunications equipment energy efficiency rating (TEEER), which is the log of TEER [PBRR11]. An overview of these metrics with a brief description is available in Table 2.2.

Network-level metrics evaluate the network energy efficiency as a whole, considering the equipment and network state, like capacity and coverage. 3GPP defines four network-level metrics, considering data volume, coverage, and the network location [3GPP20], [3GPP10]. General network energy efficiency metrics are defined as the ratio of data volume or coverage area to energy consumed by the network ($EE_{MN,DV}$, $EE_{MV,CoA}$, respectively). These are useful metrics to take into consideration while analyzing any mobile network, in particular 4G and 5G ones. The ratio of coverage area to energy consumed by the network is especially relevant to analyzing the performance of rural areas networks, as traffic demands are mostly low over larger areas. A metric aimed for urban area network evaluation is defined as the ratio of subscribed users on average during a busy hour to the power consumed at the site, as urban areas tend to have high bursts of traffic during the day [CKYY10], [HBVB11]. The inverse of $EE_{MN, DV}$, energy per unit of traffic is also widely used.

Other more situational and specific metrics are energy per connection, energy per cell site, and energy per revenue [KHSM21]. These metrics are better suited to some specific analysis as they are very

situational and cannot be used to compare networks with different topologies or objectives. These metrics are also present in Table 2.2.

Table 2.2 - Equipment and Network- level energy efficiency metrics.

Metric	Type	Units	Description
ECR	Equipment-level	Watt / Gbps	Ratio of energy consumption over effective system capacity
ECRW	Equipment-level	Watt / Gbps	Same as ECR, with power consumption calculated considering states of lower energy consumption
WCR-VL	Equipment-level	Watt / Gbps	Same as ECR, but considering specific utilization weights
ECR-EX	Equipment-level	Watt / Gbps	Same as ECR, but with extended energy saving capabilities enabled
EER	Equipment-level	Gbps / Watt	Inverse of ECR
Power per User	Equipment-level	Watt / Users	Average power consumed by each subscribed user
TEER	Equipment-level	Gbps / Watt	Ratio of “useful work” to power consumption
TEEER	Equipment-level	-Log (Gbps / Watt)	Log of TEER
$EE_{MN, DV}$	Network-level	Bit / Joule	Ratio of data volume to energy consumed
$EE_{MV, CoA}$ / PI_{rural}	Network-level	km^2 / Watt	Ratio of area covered to energy consumed
PI_{urban}	Network-level	Users / Watt	Ratio of average users during busy hours to energy consumed
Energy per Unit of Traffic	Network-level	Watt / Gb	Amount of energy needed to transmit 1 GB of data
Energy per Connection	Network-level	kW / Connection	Amount of energy needed per connection
Energy per Cell Site	Network-level	MW / cell site	Amount of energy Needed per cell site
Energy per revenue	Network-level	MW / € million	Amount of energy Needed per monetary unit

2.3 5G Services and Applications

5G service definition and differentiation is one of its major benefits in relation to past mobile networks, as it provides robust support to a diverse array of services, each with different network QoS, leading to better user experience and enhanced network optimization and performance overall.

Services in 5G can be categorized into four different classes according to [3GPP02]:

- Conversational: This class is designed for applications such as telephony speech and video

conferencing. It prioritizes very delay-sensitive traffic, ensuring low transfer time and preserving the time relation in real-time conversation. These services require low latency.

- Streaming: Used for real-time video and audio streaming, this class focuses on preserving the time relation between information entities within a flow. While it does not mandate low transfer delays, it ensures limited delay variation for time-aligned streams.
- Interactive: Applications like web browsing and server access fall into this class. It follows a request-response pattern, emphasizing round-trip delay time and transparent transfer of packet content with a low bit-error rate.
- Background: Tailored for background tasks like email delivery, this class is delivery time insensitive. It preserves packet content with a low bit-error rate, catering to scenarios where the destination does not expect data within a specific time frame.

These services are also categorized, differently, by ITU-R [ITUR15], NGMN [NGMN15] and 3GPP [3GPP16], into different categories, based on specific usage scenarios. The three main categories defined are:

- Enhanced Mobile Broadband (eMBB): eMBB services are human-centric services like the access to multimedia content, services, and data. The continuously growing demand for mobile broadband drives the development of eMBB. This category covers a wide range of services, including hotspots and wide-area coverage. Hotspots are areas with high user density and require very high traffic capacity, while having a low requirement for mobility. On the other hand, wide-area coverage requires seamless coverage and higher mobility, with high data-rates also being required. This service category is categorized by high data-rates, mobility, and coverage, while not demanding very low latency.
- Massive Machine Type Communications (mMTC): This category involves a vast number of connected devices that transmit small amounts of non-delay-sensitive data. This category is therefore defined for the necessity of connecting a large number of devices in a given area with a lower need for high data-rates.
- Ultra-Reliable and Low Latency Communications (URLLC): This category demands high-performance capabilities, including data-rates, low latency, and high availability. Examples include wireless control in industrial processes, remote medical surgery, smart grid distribution automation, and transportation safety.

2.4 Machine Learning

This section studies the main aspects of machine learning types and algorithms, focusing mainly on those that are relevant for the development of this thesis.

2.4.1 Machine Learning Types

This section is based on [DSNK17], [Mahe20], [Marq22], [PIRP22], [JKHK19]. Machine learning (ML) types fall into four distinct categories, each defined by the nature of the problem they address and the type of available data: supervised and unsupervised.

Supervised Learning

Supervised learning is the task of inferring a function that maps the inputs to the outputs of a training dataset that contains labelled data as multiple input-output pairs. The goal of this learning type is to use the training dataset to correctly and robustly predict or classify with minimum errors in new or unseen data. To do this, the training dataset is divided into a train and a test set. The chosen algorithm will use the train portion to map the input features to the outputs and use the test portion to evaluate the model. This division is done to ensure that the model evaluation is done with data the model did not train with, ensuring an independent and "true" evaluation and preventing overfitting, where the model scores very well on previously trained data but badly on new one. Figure 2.9 shows the usual supervised learning workflow.

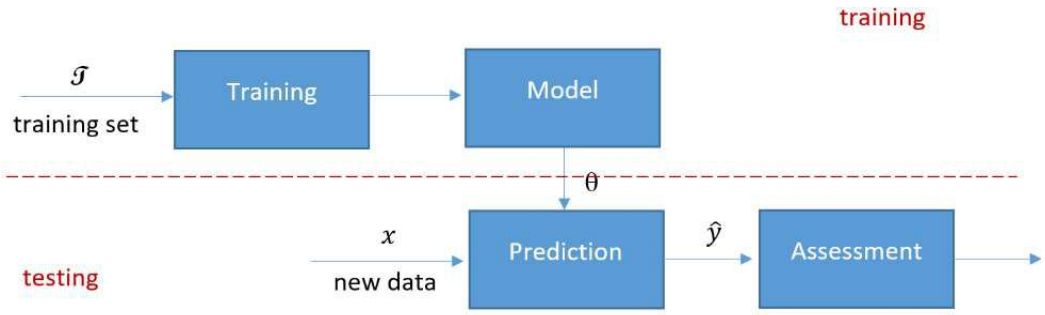


Figure 2.9 - Supervised learning workflow (extracted from [Marq22]).

During training, the model is provided with labelled data, allowing it to learn the relationships between input features and target labels where the parameters of the model are adjusted using optimization algorithms, which iteratively minimize the loss function that quantifies the prediction error. The specific method of adjustment can vary between algorithms.

Supervised learning can tackle problems of regression or classification, depending on the problem or data, where if the output is continuous the problem is a regression problem, and if the output is a label, then there is a classification problem.

To evaluate the predictions of the model, multiple performance metrics can be used. In this thesis the focus is on the most relevant ones for our work: regression performance metrics. Some popular metrics used are mean squared error Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE) and R^2. These can be defined as follows [HLGC24]:

$$\bar{\Sigma}_{MAE} = \frac{1}{n}\sum_{i=0}^{n}|(y_i - y^*)| \quad , \tag{2.2}$$

$$\bar{\Sigma}_{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y_i - y^*)^2 \quad , \tag{2.3}$$

$$\bar{\Sigma}_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - y^*)^2} \quad , \tag{2.4}$$

$$R^2 = 1 - \frac{\sum_{i=0}^{n}(y_i - y^*)^2}{\sum_{i=0}^{n}(y_i - \overline{y^*})^2} \quad , \tag{2.5}$$

where:

- n: number of observations,
- y_i: actual value,
- y^*: predicted value,
- $\bar{y}$: mean of the actual values.

Some supervised learning algorithm examples that will be explored for different applications in this thesis are neural networks, random forest and extreme random boost.

Unsupervised Learning

Contrary to supervised learning, unsupervised learning is the task of inferring a function from unlabeled data to depict some structure or attributes on its own by learning some features from the data itself. These algorithms are usually used for clustering, feature reduction, pattern recognition and statistics.

These tasks can be grouped into clustering and association tasks, where the first aims to discover the inherent groupings in the data and the latter to discover relationships or patterns withing the data and to identify associations or connections between different elements.

The k-means algorithm is an example of unsupervised learning used in this thesis.

2.4.2 Machine Learning Algorithms

In the context of this work, some ML algorithms stand out in relation to others regarding the ability to help solve the identified problem, most notoriously supervised and unsupervised learning algorithms as they leverage labelled and unlabeled data. In this section, the details and functionality of these algorithms are explained.

K-Means Clustering

The k-means algorithm is a widely used unsupervised learning method for clustering data into a predefined number of distinct groups based on feature similarity. It is particularly effective when the goal is to partition a dataset into some clusters, where each data point belongs to the cluster with the nearest centroid, or cluster center, which is the central point of a cluster, representing the mean position of all data points within that cluster [KTMP13].

This algorithm operates through a series of iterative steps that seek to minimize the overall variance within clusters, called inertia. The process begins with the operator specifying the number of clusters, *k*,

and selecting *k* random initial centroids from the dataset. The Euclidean distance between the different data points and the cluster centroids is calculated, assigning each datapoint to the cluster with the least Euclidian distance from itself. The Euclidean distance is the straight-line distance between two points in Euclidean space, calculated using the Pythagorean theorem. It can be calculated using (2.6) [LLLC10].

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.6}$$

where x_i and y_i are the coordinates of the points in each dimension.

Once the data points have been assigned to clusters, the algorithm recalculates the centroids by averaging the points within each cluster. This update step shifts the centroid towards the mean of the assigned points, improving the representation of the cluster. The key objective of the k-means algorithm is to minimize the inertia, a measure of within-cluster variance. Inertia is defined as the sum of squared distances between each data point and its corresponding centroid [KTMP13]:

$$I_{IN} = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - C_j\|^2 \ , \tag{2.7}$$

where:

- x_i: coordinates of the points,
- C_j: coordinates of each j clusters centroids.

In simpler terms, inertia quantifies how tightly the data points are grouped around their centroids. Lower inertia means that data points are closer to their cluster centroids, indicating better-defined clusters. Minimizing inertia is important because it ensures that the resulting clusters are compact and separated. The algorithm repeats the assignment and update steps until the centroids stabilize, meaning they no longer move significantly, or until a predefined number of iterations is reached [KTMP13].

Artificial Neural Network

An artificial neural network (ANN) takes inspiration on a biologic one, composed of multiple “neurons” interconnected to form a network capable of solving complex problems [ICSJ19].

A neuron is the basic node in an ANN and its job is to perform computations to the signals it receives and propagate its results to other neurons. These computations are functions, called activation functions, where the sum of inputs is passed through it and results in a scalar output. Some popular activation functions used are [SSSS20]:

- ReLU: ReLU stands for rectified liner unit and is a non-linear activation function which is widely used in neural networks. It transforms values in the range of 0 to 1, where negative input values are mapped to 0, as positive ones stay the same. The upper hand of using ReLU function is that all the neurons are not activated at the same time, meaning that the neuron is deactivated if the output is 0. It can be defined as:

$$U_{ReLu}(x) = max(0, x) \quad , \tag{2.8}$$

Figure 2.10 shows a plot of this function.

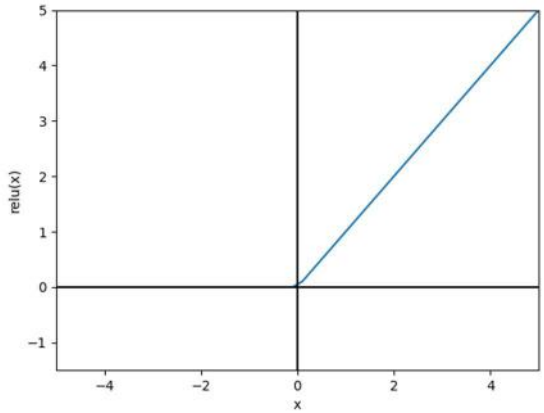


Figure 2.10 - ReLU Function.

- Sigmoid: One of the most used activation functions, as it is a non-linear function. It transforms the values in the range 0 to 1. It can be defined as:

$$sigmoid(\mathrm{x}) = \frac{1}{e^{-x}} \quad , \tag{2.9}$$

Figure 2.11 shows a plot of this function.

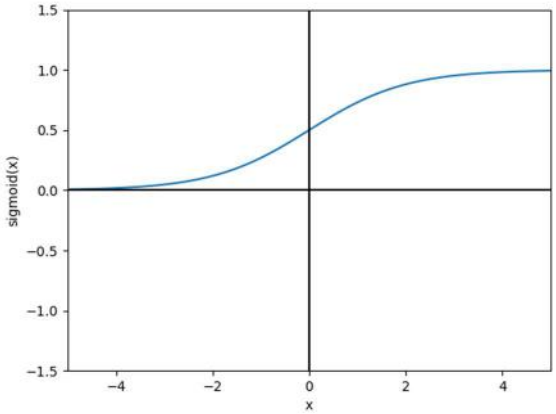


Figure 2.11 - Sigmoid Function.

- Tanh: It is the hyperbolic tangent function. It is similar to the sigmoid function but is symmetric around the origin, transforming values into the range of -1 to 1. This function has a steeper gradient (how much the output of the function changes with respect to changes in the input) meaning that closer input values will result in more far apart output ones, which can lead to faster learning. It can be defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad , \tag{2.10}$$

Figure 2.12 shows a plot of this function.

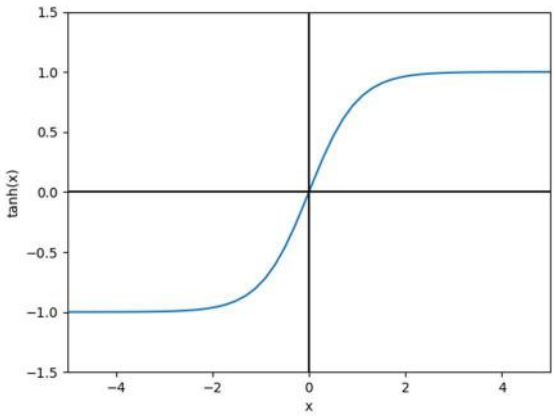


Figure 2.12 - Tanh Function.

The connections between neurons are called weights. The output of each neuron is multiplied by this weight value before being passed down, where the weight value determines the importance of output of that neuron to the final output. An ANN consists of an input layer, with the same number of neurons as the number of inputs, an output layer with the number of neurons equal to the number of possible outputs and can have some middle layers with several neurons depending on the application [ICSJ19], [DKAK12], [BKPS93].

There are many different types of ANN, categorized based on their architecture. Some of these, particularly those most relevant to this work, are explained below [ICSJ19], [Sher20]:

- Feed Forward ANN: One of the simplest types of ANN. In feed forward ANNs, data is passed in only one direction, from the input neurons, through middle layers if they exist and ending in the output neurons. A common type of feed forward ANN is a multi-layer perceptron (MLP), where there is an input layer of neurons, one or more middle layers, called the hidden layers, and a final output layer. When there are many hidden layers present in an MLP, Feed forward ANNs are used for complex classification, regression, pattern recognition, optimization, function approximation and other more specific use cases. Figure 2.13 illustrates the architecture of a possible feed forward ANN.
- Deep Neural Network (DNN): A DNN is a type of ANN that consists of many hidden layers, sometimes reaching tens of layers deep. This extensive layering allows DNNs to capture and model highly complex patterns and hierarchical features in the data. The deeper structure enables more nuanced understanding and representation, making DNNs highly effective for tasks involving large datasets and intricate relationships, such as predictive modeling. The increased depth offers superior performance over traditional, shallower networks by leveraging the added complexity to achieve greater accuracy and flexibility.
- Long Short-Term Memory (LSTM) –LSTM is a type of Recurrent Neural Network (RNN). RNNs work similarly to feedforward ANNs, but with the added feature that each neuron maintains a

hidden state to keep track of information from previous time steps. This way, the RNN can remember past states and easily retain sequential patterns present in the data. One problem in RNNs is that the model has a hard time learning long-term dependencies, which is crucial for time series prediction. Advanced architectures like LSTM networks were developed to mitigate this issue as they are able to preserve information over longer sequences. LSTMs incorporate a memory cell that allows them to maintain information over long periods compared with the hidden states from RNNs. Additionally, LSTMs use three types of gates to manage information: the input gate, which controls how much new information from the current input is added to the memory cell; the forget gate, which decides which parts of the existing memory cell should be discarded or retained; and the output gate, which determines what information from the memory cell should be output and used to compute the prediction for the next time step. These gates collectively enable LSTMs to manage long-term dependencies more effectively than standard RNNs. LSTMs are especially good in handling time series predictions, like BS traffic or energy consumption data, which is useful for our problem.

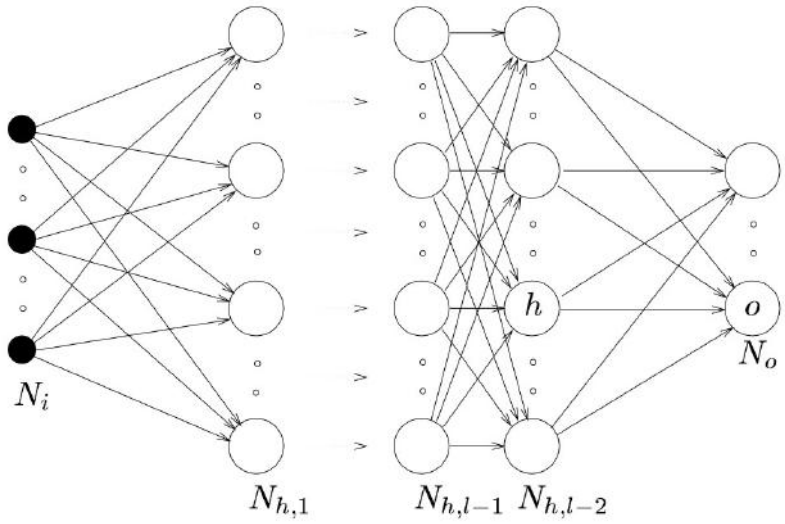


Figure 2.13 - Example of a multi-layer ANN architecture (extracted from [BKPS93]).

ANNs can be integrated to perform all types of learning types stated in Section 2.4.1, making them very flexible to perform a wide range of tasks.

In ANN, learning is the process of changing the weights of connections to optimize the output according to the problem [DKAK12]. In supervised learning such is accomplished by using the backpropagation algorithm. In this algorithm, weights are initially initialized with random values. When training begins, each training sample is passed through the ANN and the error between the output and the expected value is calculated using an error function. This error is then propagated from the output layers until the input layers, where the weights are adjusted to minimize this error function using an optimization algorithm. This process is repeated for all samples of the training dataset [DKAK12], [BKPS93]. An optimization algorithm is a method used to adjust the weights of a model to minimize the error function, thereby enhancing the performance of the model. There are multiple optimization algorithms that can be used with ANNs. One popular optimizer for time series prediction is the Adam optimizer.

When creating ANNs, the number of hidden layers is an important parameter to its performance. Depending on the problem, a specific number of hidden layers is optimal, where less and more than that number will hurt the ANNs performance by causing underfitting or overfitting, respectively. While increasing the number of hidden layers will always lead to less error on the training dataset, this will cause performance to drop in a real scenario on untrained data. Adding more hidden layers also increases the number of parameters, with the downside of making training more difficult [BKPS93]. This is shown in Figure 2.14.

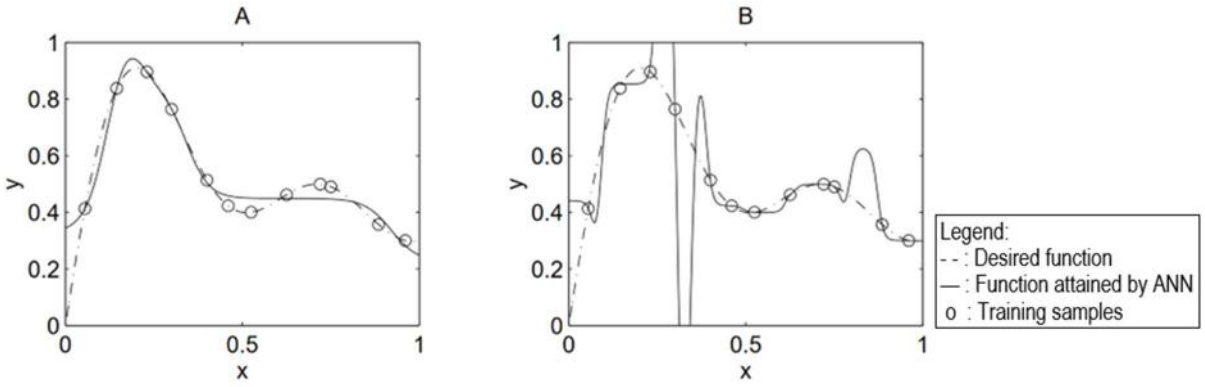


Figure 2.14 - Effect of the number of hidden layers on the network performance. A) 5 hidden layers, B) 20 hidden layers (extracted from [BKPS93]).

This is a crucial aspect of defining an ANN model. To achieve optimal results, it is essential to explore and fine-tune the various parameters available. This process involves systematically searching through the possible configurations, including the number of layers, neurons per layer, activation functions, etc. By carefully adjusting these parameters, one can enhance the performance of the model and improve its ability to generalize effectively to new, unseen data.

Another critical point is the data the ANN is training on. This dataset must contain enough diverse data that represents the whole system. If this does not happen, there is no guarantee that the ANN will perform well, as there will not be enough information for it to properly learn [BKPS93]. This is depicted in Figure 2.15.

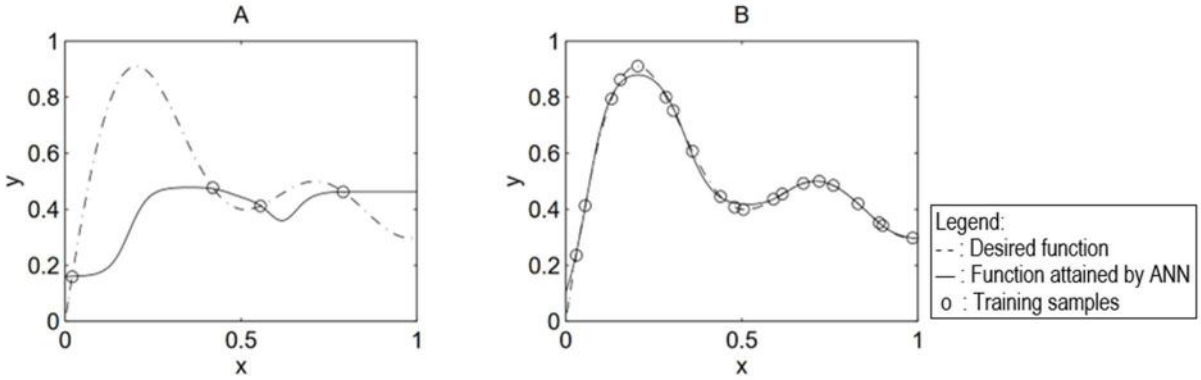


Figure 2.15 - Effect of the learning set size on the network performance. A) 4 learning samples, B) 20 learning samples (extracted from [BKPS93]).

2.4.3 Feature Selection

When working on a dataset and applying machine learning techniques, feature importance becomes a critical consideration. The “Curse of Dimensionality” as described by Bellman, refers to the challenges associated with high-dimensional data. Higher dimension data often contains noisy, irrelevant, and redundant features, which can lead to model overfitting and increased error rates in learning algorithms. To address these issues, dimensionality reduction techniques are applied as part of the preprocessing stage [BVJA19]. These techniques help to simplify models and improve performance by focusing on the most relevant features and removing irrelevant and redundant ones.

Feature selection methods can be classified into 3 categories: Filter, Wrapper and Embedded. Filter methods are independent of the learning algorithm, where features are selected based on statistical measurements, making them simpler computationally wise. Mutual information and correlation coefficients are some examples of filter methods. Wrapper methods use different features and feature subsets and compare the used model´s performance, selecting the best subset based on model performance. This method is more accurate than the filter method, but computationally more expensive. Recursive feature elimination and sequential feature selection are some examples of wrapper methods. Embedded methods use ensemble and hybrid learning methods, where feature selection is performed as part of the model training process. The selection of features is integrated with the learning algorithm itself. Random forest and extreme gradient boosting are some examples of embedded methods [BVJA19], [LCWM10].

Mutual Information

Mutual Information is a measure of the amount of information that one variable contains about another variable, meaning how much knowing about one variable reduces the uncertainty of the other. Equation (2.11) shows how mutual information is calculated between two continuous variables *x* and *y*, with joint Probability Density Function (PDF) $p(x, y)$, and marginal pdfs $p(x)$ and $p(y)$ [ETPZ09],

$$I(X;Y) = \iint p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy \quad . \tag{2.11}$$

The calculation of mutual information requires the estimation of pdfs or entropies from the data samples, which usually are not known. One way to do this is to estimate entropies directly from the data using k-nearest neighbors’ distances. Larger distances mean that the density is smaller and the same is true the other way around. Once densities and then entropies are calculated, mutual information can be obtained using (2.12) [AKSG08],

$$I(X;Y) = H(X) + H(Y) - H(X;Y) \quad . \tag{2.12}$$

When using mutual information for feature selection, the features that have higher mutual information with the target variable are considered to be the most important ones, that provide the best prediction of the target variable. The range of values for mutual information is a positive number, with their importance being relative to each other’s values.

Correlation Coefficients

Correlation is a method of finding the relationship between two variables. Two of the most common correlations that can be used to obtain correlation coefficients between variables in a dataset are Pearson's and Spearman's correlation. Pearson's correlation measures the linear relationship between two variables. These correlation coefficients range from -1 to 1. A positive linear correlation results in a positive coefficient and a negative one in a negative coefficient, where 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship and 0 indicates the absence of a linear relationship. Pearson's correlation coefficients can be obtained by the ratio of the covariance of the two variables to the product of their respective standard deviations, shown in (2.13) where sample covariance and sample standard deviation are plugged in [Chok10],

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - y)^2}} \ . \tag{2.13}$$

Spearman's correlation measures the monotonic relationship between two variables, even if it is a non-linear relationship. Spearman correlation calculates the correlation based on the ranks of the values instead of using the original data values like Pearson. Ranks refer to the ordinal positions of the data points within each variable. Spearman's correlation coefficients range from -1 to 1, where a positive value means a positive monotone relationship and a negative value means a negative relationship, and where -1 corresponds to a perfect negative monotone relationship and 1 corresponds to a perfect positive monotone relationship, it being linear or non-linear, and 0 corresponds to no monotonic relationship. The Spearman's correlation coefficient is calculated using (2.14) [Chok10],

$$\rho_s = \frac{\sum_i (R(x_i) - R(\bar{x}))(R(y_i) - R(\bar{y}))}{\sqrt{\sum_i (R(x_i) - R(\bar{x}))^2 \sum_i \big(R(y_i) - R(\bar{y})\big)^2}} \ , \tag{2.14}$$

where $R(x_i)$ and $R(y_i)$ are the ranks of the observation in the sample.

Random Forest

Random Forest is a mechanism versatile enough to deal with both supervised classification and regression tasks. A random forest is a predictor consisting of a collection of M randomized classification or regression trees. Each tree is trained on a different random subset of data and features originated from the original dataset, known as bootstrap samples. During training, the CART-split criterion is used to construct and split each tree. At each node of each tree, different features and thresholds are considered so that the best split is selected, which is the one that reduces the Gini impurity, an error metric used in classification tasks or mean squared error in regression tasks [BGSE15].

This training procedure is what allows random forest algorithms to calculate the importance of each feature: Mean Decrease Impurity (MDI) is calculated based on the total decrease in node impurity from splitting on the feature, averaged over all trees. This means that during training, every time a tree is split using a certain feature, the reduction in Gini impurity or mean squared error is aggregated and averaged

over all other trees. These values are then normalized so that their sum equals 1, where higher values indicate higher feature importance, and can also be called Gini Importance [BGSE15].

Another feature importance measurement is Mean Decrease Accuracy (MDA). MDA is based on the idea that if a feature is not important, shuffling its values should not degrade prediction accuracy. MDA is obtained by averaging the difference in error estimation before and after the permutation over all trees. This can also be called permutation importance [BGSE15].

Extreme Gradient Boosting (XGBoost)

XGBoost is a scalable and efficient implementation of gradient boosting machines. The algorithm uses decision trees as base learners. Each tree is added to the model to minimize a loss function, focusing on the residual errors of the prior trees where the final model is an ensemble of all the trees [CTHT24].

Similarly to random forests, XGBoost calculates feature importance with the results from its training. Five possible feature importance metrics can be obtained using XGBoost [CTHT24]:

- Weight: Weight, also known as frequency, is the number of times a feature appears in the trees of the model. It provides a count of how often a feature is used to make decisions in the branches.
- Gain: Gain measures the average improvement in accuracy that a feature provides when it is used in trees. It reflects the contribution of a feature to reducing the error of the model. A higher gain indicates a more important feature.
- Cover: Cover represents the average number of samples affected by a feature when it is used in trees. It indicates the coverage or impact of a feature on the training dataset. Features that influence a larger number of samples are considered more important.
- Total Gain: Total gain is the sum of gains for a feature across all trees. It shows the overall contribution of a feature to the accuracy improvement of the model.
- Total Cover: Total cover is the sum of coverage for a feature across all trees. It provides the total impact of a feature on the dataset throughout the entire model.

Sequential Feature Selection

Sequential Feature Selection (SFS) is a greedy algorithm that iteratively adds or removes features from a dataset in order to improve the performance of a predictive model. SFS can be either Forward Feature Selection (FFS) or Backward Feature Elimination (BFE), where features are sequentially added or removed from the model.

When doing forward feature selection, the model starts with an empty feature set. The model is then fit using each feature, and the one that gives a better improvement in prediction is chosen. The same task is repeated until the feature set is complete. When backward selection is chosen, the opposite happens. The model is first fit with the full feature set. After that, the model is fit with one feature missing, doing that for all features once. The feature which least impacts prediction score is removed, repeating this process until arriving at the desired feature set length.

Sequential Feature selection is good at removing redundant features, even if they are highly correlated with the target variable, since redundant features may carry a significant amount of the same predictive information, once the more relevant one is introduced, the other one will not produce a meaningful change in the error rate, leading to it not getting picked [Brei01].

2.5 State of the Art

In this section, we review previous work on time-based energy efficiency techniques, specifically the use of sleep modes in 4G/5G networks. The goal is to examine existing solutions, focusing on methodologies and approaches to enhance energy efficiency, establishing a foundation for our proposed work and identifying gaps in the current literature.

The authors in [PPDG22] develop a comprehensive power consumption model for realistic 5G base stations (BS). They present an ANN model based on extensive data from a 5G BS deployment in China, achieving high accuracy in estimating power consumption under various conditions. This methodology is useful since predicting BS energy consumption is crucial for obtaining feedback on attempts to lower network energy consumption. Our work will build upon this by integrating these predictions into the final proposed algorithm.

In [SGAC17], the authors propose an approach for implementing Autonomous Sleep Modes (ASMs) in 5G networks to enhance energy efficiency. This technique buffers user service requests during idle periods of BSs and activates ASMs to reduce power consumption. The study focuses on connected mode users and considers various signaling periodicities. The results show significant reductions in energy consumption, reaching up to 90% for low loads, by increasing signaling periodicities; however, this comes at the cost of throughput degradation (up to 19%) and increased latency (up to 5 ms). This paper is highly relevant to our thesis, as the proposed algorithm builds upon the concept of autonomous sleep modes. In our case, these sleep modes will be activated based on traffic predictions, making the approach more dynamic and responsive to real-time network conditions. By incorporating traffic forecasts, our method offers a more adaptive and context-aware solution, ensuring that energy-saving actions are taken with minimal impact on network performance.

The work [SCAG19] explores optimal control strategies for implementing ASMs in 5G NSA networks, focusing on the trade-off between energy consumption reduction and delay incurred by sleep modes. The study utilizes Markov Decision Processes to derive different policies based on the priorities set by the network operator regarding delay and energy savings. The results demonstrate how varying the weights assigned to delay, energy reduction, and switching costs influences system performance. This work suggests potential enhancements for various use cases, such as URLLC, and considers future extensions to include signaling periodicity in decision processes for 5G SA architecture. This paper outlines a methodology similar to the proposed algorithm in this thesis, with the key difference being that our approach does not rely on reinforcement learning. Instead, due to the nature of the available data, we leverage predictions made with an LSTM network to guide the actions of the algorithm.

Chapter 3

Development

This chapter outlines the development of the energy-saving algorithm. It details the methodologies, processes, and techniques employed in creating the algorithm, highlighting the key components that contribute to its effectiveness. This chapter lays the groundwork for subsequent analyses and evaluations.

3.1 Implementation Outline

To achieve significant energy savings in 5G networks using ML, an algorithm was designed that not only delivers substantial energy efficiency but is also customizable to meet the specific requirements of network operators while remaining easy to implement and maintain. The objective of this algorithm is to reduce energy consumption by identifying periods of low or zero network utilization and taking appropriate actions to optimize energy efficiency during these times.

The algorithm can be broken down into small functions, as represented in Figure 3.1.

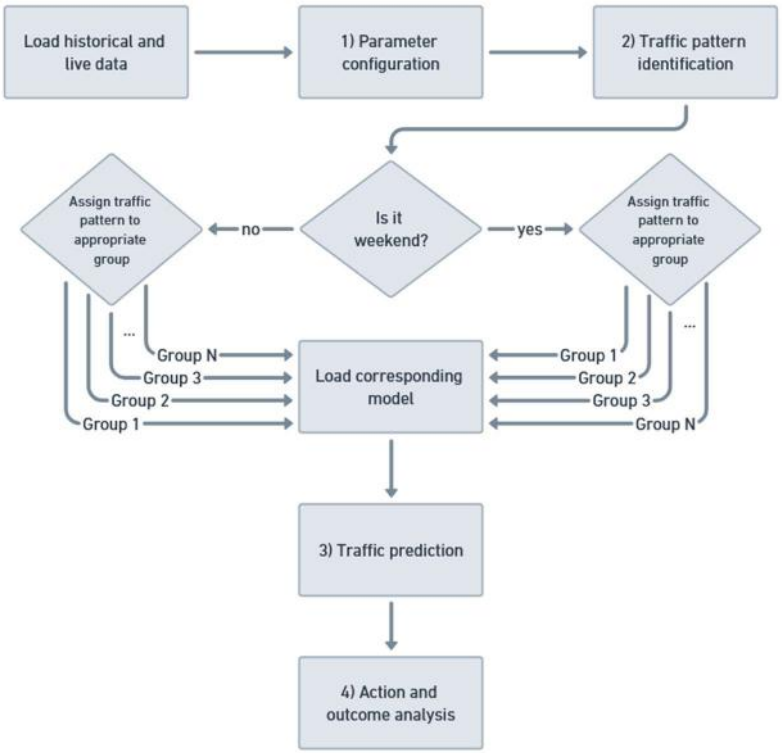


Figure 3.1 – Energy saving algorithm flowchart.

The different functions presented in Figure 3.1 can be defined as follows:

1) Parameter configuration: The operator defines PRB usage thresholds in percentage for the model to act upon. This PRB usage threshold dictates when the load is low enough for the energy saving action to happen. Additionally, operators define a confidence metric ranging from 0 to 1, which adds a confidence interval to the predictions of the model, making them either more aggressive or passive.
2) Traffic pattern identification: To obtain the best results, the average daily PRB usage is computed from historical data so that the appropriate model is chosen for prediction. Since weekday and weekend patterns are very distinct, these are treated differently, meaning that both have their separate pattern computed and model chosen.
3) Traffic prediction: Using the data available, multiple LSTM models were trained, one for each traffic pattern present in the data. While the algorithm is running, predictions for the next time

step are made over time using the appropriate LSTM models.

4) Action and outcome analysis: Based on the thresholds, when a prediction is made at each time step, an action follows. The outcome of the action is displayed to the operator, enabling the evaluation of the performance of the algorithm and the ability to make adjustments if necessary.

Following these steps, a “plug and play” type of algorithms is created, where an operator would simply need to provide the algorithm with historical traffic data of the sector as well as the wanted energy saving parameters, to work with our program, making it a simple and flexible approach to saving energy to an operator.

3.2 Dataset Details

A dataset from Vodafone of real BS traffic and radio performance was received, with the goal of understanding how the radio usage and traffic relate to the energy consumption and finding possible energy saving opportunities available in the data.

The datasets contain data from two countries, which throughout this thesis will remain confidential and will be referred to as country A and country B. For each country, the dataset contains data from individual BSs, each belonging to a cluster of BSs. A BS cluster is a group of BSs that have similar characteristics and are geographically located close to each other. Additionally, each country BS antennas are provided by different manufacturers, also kept confidential, which explains the variations in the datasets, including differences in the available counters, network performance metrics collected by base stations, and time granularities. The country A dataset has data of 1 month from 4 clusters, each of 4 BS with 3 sectors each using the C-band range band with a bandwidth varying from 80 to 90 MHz and 30 kHz SCS. The country B dataset has data of 3 months from 3 clusters, each of 2 BS with 3 sectors each using the C-band range band with a bandwidth of 40 MHz and 30 kHz SCS. All sectors are equipped with 32/32 or 64/64 RX/TX MIMO systems. The locations of these BSs are mostly urban residential areas, with occasionally different infrastructures like shopping centers, hospitals, and public leisure places. The detailed locations and characteristics of each BS and sector and the datasets can be found in Annex A.

Each dataset contains counter data from radio usage, energy usage and quality of service (QoS) and signal to interference plus noise ratio (SINR) value distributions of 15-minute time intervals in case of country A dataset and 1 hour in case of country B. Both datasets have data from all BS mixed. To facilitate interpretation and usage, both are separated into multiple Excel files, one for each BS present, and each sector is also separated in different sheets of the corresponding Excel file.

Table 3.1 provides a comparison of the datasets from each country, highlighting key differences such as the number of clusters, sectors, time granularity, and bandwidth. These variations can be attributed, in part, to the use of antennas from different manufacturers in each country, which results in differences in data collection and structure.

Table 3.1 – Comparison of the characteristics of the datasets across countries.

Characteristic	Country A Dataset	Country B Dataset
Number of Clusters	4 clusters	3 clusters
Number of BS	16 BS (4 per cluster)	6 BS (2 per cluster)
Number of Sectors	48 sectors (3 per BS)	18 sectors (3 per BS)
Time Period	1 month	3 months
Time Granularity	15 minutes	1 hour

The detailed presentation and explanation of the counters present in country A dataset are provided in Table 3.2. The real counter names are exchanged for representative names in order to keep the confidentiality of the antenna manufacturers.

Table 3.2 – Country A dataset counter description.

Counter Name	Symbol	Unit	Description
N_UE_DL	$N_{activeUE}^{DL}$	-	Cumulative count of active User Equipment instances per slot that have data stored in the downlink buffer.
Max_N_UE_DL	$Max_{activeUE}^{DL}$	-	Peak number of active UE instances DL data from a 1 second time sampling.
N_UE_UL	$N_{activeUE}^{DL}$	-	Cumulative count of active User Equipment instances per slot that have data stored in the uplink buffer.
Max_N_UE_UL	$Max_{activeUE}^{UL}$	-	Peak number of active UE instances UL data from a 1 second time sampling.
N_Rbsym_typeA	N_{RBsym_A}	-	Count of resource block symbols used for DL mapping type A. Includes symbols used for demodulation reference signals and phase-tracking reference signals. A resource block symbol is defined as a subcarrier in frequency and a slot in time.
N_Rbsym_Broadcast	N_{RBsym_B}	-	Count of resource block symbols used for downlink mapping type A broadcasting. Includes Synchronization Signal, Physical

			Broadcast Channel, Broadcast Control Channel, and Paging Control Channel including System Information Blocks and Random-Access Msg2.
N_Rbsym_Signaling	N_{RBsym_CSI}	-	Count of resource block symbols utilized for Channel State Information Reference Signal, Transmit Reference Signal, and CSI Interference Measurement.
N_Rbsym_available	N_{RBsym_Total}	-	Count of RB symbols available for DL transmission.
N_PRB_available	$N_{PRB_available}$	-	Number of available PRBs for DL transmission.
T_DL	T_{DL}	Each counter step resembles 125 µs.	Total time spent on scheduling the initial transmission of DL channels user data. Does not count retransmissions.
Vol_DL_Data	V_{DL}	Byte	Total volume of DL data transmitted for the initial transmission, for each channel, of user data. Excludes retransmissions.
T_DL_active	T_{DL_active}	Each counter step resembles 125 µs.	Total duration of slots during which there is DL scheduling activity. Stepped for each slot where there is at least one Signalling Radio Bearer (SRB) or Data Radio Bearer (DRB) PDSCH scheduling activity. Other scheduling activities (Msg2 or Msg4, paging and so on) are excluded.
Vol_DL_Data&Signaling	$V_{DL\_Data\&Signaling}$	Byte	Total volume of transmitted DL data, including both user data (DRB) and control signaling (SRB), measured in the MAC layer.
N_RCC_Users_SA_Max	Max_{RCC_SA}	-	Maximum number of RRC connected users in SA mode, read per 5 seconds
N_RCC_Users_NSA_Max	Max_{RCC_NSA}	-	Maximum number of RRC connected users in en-DC mode, read per 5 seconds
CQI_Rank_1-4	$R1-4_{CQI}$	-	Distribution of UE-reported CQI values related to rank 1-4 transmissions when highest modulation order supported in DL is configured to be 256-QAM.
SINR_PUCCH	$SINR_{PUCCH}$	-	Distribution of SINR values calculated for Physical Uplink Control Channel.

SINR_PUSCH	$SINR_{PUSCH}$	-	Distribution of SINR values calculated for Physical Uplink Shared Channel.
Radio_Energy	E_{Radio}	Wh	Sum of the energy consumption of radio units.
BBU_Energy	E_{BBU}	Wh	Sum of the energy consumption of Base Band Units (BBU).

The same is done for country B dataset in Table 3.3.

Table 3.3 – Country B dataset counter data description.

Counter Name	**Symbol**	**Unit**	**Description**
N_UE_DL	$N^{DL}_{activeUE}$	-	Cumulative count of active User Equipment instances per slot that have data stored in the downlink buffer.
N_UE_UL	$N^{UL}_{activeUE}$	-	Cumulative count of active User Equipment instances per slot that have data stored in the uplink buffer.
N_UE_DL_Avg	$\bar{N}^{DL}_{activeUE}$	-	Average number of active User Equipment instances that are scheduled in the downlink buffer. Sampled per second.
N_UE_UL_Avg	$\bar{N}^{UL}_{activeUE}$	-	Average number of active User Equipment instances that are scheduled in the uplink buffer. Sampled per second.
N_RCC_Users_Avg	$\bar{N}_{RCC\_users}$	-	Average number of RRC connected users, sampled per second.
N_RCC_Users_Max	Max_{RCC_users}	-	Maximum number of RRC connected users, sampled per second.
N_PRB_Used_AVG	$\bar{N}_{PRB\_used}$	-	Average number of PRBs used in DL transmissions.
N_PRB_available	$N_{PRB\_available}$	-	Number of available PRBs for DL transmission.
T_DL_Rmv_LastSlot	$T_{DL_{RmvLastSlot}}$	µs	Data DL transmission duration in microseconds. The tail packet transmission is not counted.
Vol_DL_Data	V_{DL}	kbit	Total downlink traffic volume in a cell in kbits, measured at the RLC layer.
Vol_DL_Data_LastSlot	$V_{DL_{LastSlot}}$	kbit	Total downlink traffic volume in a cell in kbits, measured at the RLC layer. Only the tail packet transmission is counted.

Vol_DL_Cell	$V_{DL_{cell}}$	kbit	Total downlink traffic volume in a cell in kbits, measured at the MAC layer.
T_DL_Cell	$T_{DL_{cell}}$	µs	Data DL transmission duration in microseconds of the cell.
SINR_PUCCH	$SINR_{PUCCH}$	-	Distribution of SINR values calculated for Physical Uplink Control Channel.
SINR_PUSCH	$SINR_{PUSCH}$	-	Distribution of SINR values calculated for Physical Uplink Shared Channel.
RU_Power_Avg	$\bar{P}_{RU}$	dBm	Average power consumption of radio units, measured in dBm and sampled every 5 seconds.
RU_Power_Max	P_{RU_Max}	dBm	Maximum value of the power consumption of radio units, measured in dBm and sampled every 5 seconds.
BBU_Energy	E_{BBU}	kWh	Sum of the energy consumption of the BBUs. Measured in kWh.
BTS_Energy	E_{BTS}	kWh	Total energy consumption of the Base Transceiver Station (BTS), encompassing Remote Radio Units (RRUs) or Active Antenna Units (AAUs).

Both datasets come with a list of traffic metrics that can be formulated from these counters and will be important in the analysis and relations between traffic and energy consumption, as well as to obtain key information that will help us develop better suited energy-saving techniques. These traffic metrics are crucial because they reflect traffic and cell usage patterns, which are key for understanding the correlations between network activity and energy consumption. By analyzing these metrics, opportunities to optimize energy use can be identified. These traffic metrics are listed in Table 3.4, along with the formulas used to obtain them.

Table 3.4 - Performance metrics formulated from the dataset parameters.

Performance Metric	**Symbol**	**Country A Formula**	**Country B Formula**
N_PRB_Used_AVG	$\bar{N}_{PRB\_used}$	$\frac{N_{RBsym_A} + N_{RBsym_B} + N_{RBsym_{CSI}}}{N_{RBsym_total}} \times N_{PRB_{available}}$	Present in the dataset
THP_User_NonGBR [Mbps]	Thp_{DL}	$\frac{64 \times V_{DL\_Data\&Signaling}}{T_{DL} \times 1000}$	$\frac{1000 \times (V_{DL} - V_{Dl_LastSlot})}{T_{DL_{RmvLastSlot}}}$
Vol_DL_MAC [MB]	V_{DL_MAC}	$\frac{V_{DL\_Data\&Signaling}}{10^6}$	$\frac{V_{DL_{Cell}}}{8 \times 10^3}$
T_DL_Active_s [s]	$T_{DL\_active\_s}$	$\frac{T_{DL_{Active}}}{8 \times 10^3}$	$\frac{T_{DL_{Cell}}}{10^6}$

After further inspection and deliberation, two problems were identified with the datasets, particularly as they lacked the ideal types of data necessary to support our intended implementation of energy-saving techniques in the network. The two main problems with the data are:

1. **Time Granularity:** One of the major issues with the dataset is the coarse time granularity. Initially, a much finer granularity was expected, as the goal was to develop a model capable of identifying periods of zero traffic and turning off the station during these idle times to conserve energy. However, with 15-minute or 1-hour time steps, it becomes nearly impossible to pinpoint these moments of no traffic. A lot can happen within a 15-minute window, which means that potential idle periods are masked by aggregated data. As a result, the original approach needs to be reconsidered in favor of finding alternative routes to energy savings within the network. Additionally, the lack of fine granularity presents a challenge for predictive modeling. With larger time steps, short-term fluctuations and subtle patterns in traffic are smoothed out or lost, making it difficult for the model to capture smaller variations. This limits the ability of the model to accurately anticipate rapid changes in traffic, reducing its overall precision.
2. **Insufficient Data to Support Finer Granularity Reconstruction:** Another issue with the dataset is the absence of detailed statistics during the 15-minute or 1-hour time frames. Without metrics such as the number of distinct active users, the number of transmissions, or other critical counters, it is impossible to recreate a finer-grained version of the dataset that would accurately reflect the real network scenarios. Additionally, the counters consist mostly of cumulative sums of multiple users' traffic, making it impossible to distinguish individual users. This aggregation further complicates the reconstruction of detailed traffic patterns, as it masks the behavior of individual users. This is problematic, as the goal is to remain true to the data provided by Vodafone without generating synthetic features that could deviate from reality.

While it might be technically feasible to generate a finer granularity dataset through estimates and assumptions, doing so would compromise the realism of the data, making the results less reliable, as accurately simulating such fine granularities would be extremely difficult. For the analysis to be valid and directly applicable to real-world implementations, it is essential to work with the provided counters, ensuring that the results remain grounded in actual network conditions. Despite the dataset limitations, where more detailed data could potentially lead to improved results, this work remains both practical and realistic, serving as a foundation and proof of concept for further development and optimization.

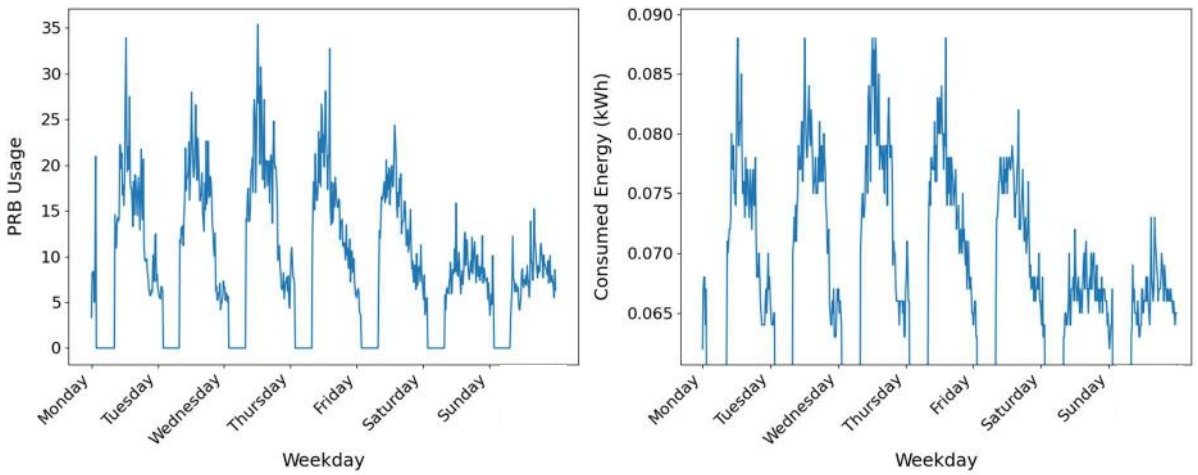
It was ultimately decided to use the datasets exactly as provided by Vodafone, as the primary objective is to propose energy-saving techniques based on the data currently available.

3.3 Exploratory Dataset Analysis

In order to fully understand the dataset and its features, an exploratory dataset analysis is conducted with the goal of uncovering patterns, relationships and the main features that influence energy consumption, providing the foundation for subsequent machine learning model definition.

3.3.1 Data Visualization and Statistics

The first step was to visualize the different features of the dataset to understand how these changed over time. Figure 3.2 shows the plots of the counters N_PRB_Used_AVG and Radio_Energy, which depicts the cell usage and power consumption of RRU of one sector across one week. It is important to note that this plot is very similar to all the other ones from other sites and sectors, which means that the observations made are valid for the whole dataset.



(a) N_PRB_Used_AVG. (b) Radio Energy.

Figure 3.2 - Plot of average used PRB and corresponding RRU consumed energy over one week.

Multiple observations can be made from Figure 3.2:

- Cell usage follows a consistent daily pattern, though this pattern varies across the different sectors depending on the site and its geographic location.
- Within the same site this daily pattern is different between weekdays and weekends.
- For the datasets of both countries, the RRU is sometimes shut down during the nighttime, approximately from 1:00 AM to 6:00 AM or 8:00 AM in the morning, depending on the country. This is not true for some sites with nighttime usage demands and, specifically for sites of country A, this does not happen for a period of one week.
- Radio energy consumption appears to follow cell usage, along with other features such as activity time and DL data volume.

With these observations in mind two more analysis were done: First, pivot tables were created to show the mean values of each feature for every weekday and corresponding hours. Second, the feature statistics of the dataset, such as the average, maximum, minimum, and standard deviation across all days, were calculated and plotted over a 24-hour period. Since the cells switch off during the night, to not affect statistics and average values, the times in which the cell is switched off were removed for this analysis.

Figure 3.3 shows a heatmap pivot table of the counter N_PRB_Used_AVG of a sector, depicting the average PRB utilization over each day of the week and each hour within each day. This visualization

helps understand how the network usage, in this case PRB usage, evolve through the day and week, as well as help observe trends in the data. The same is done for the other relevant counters.

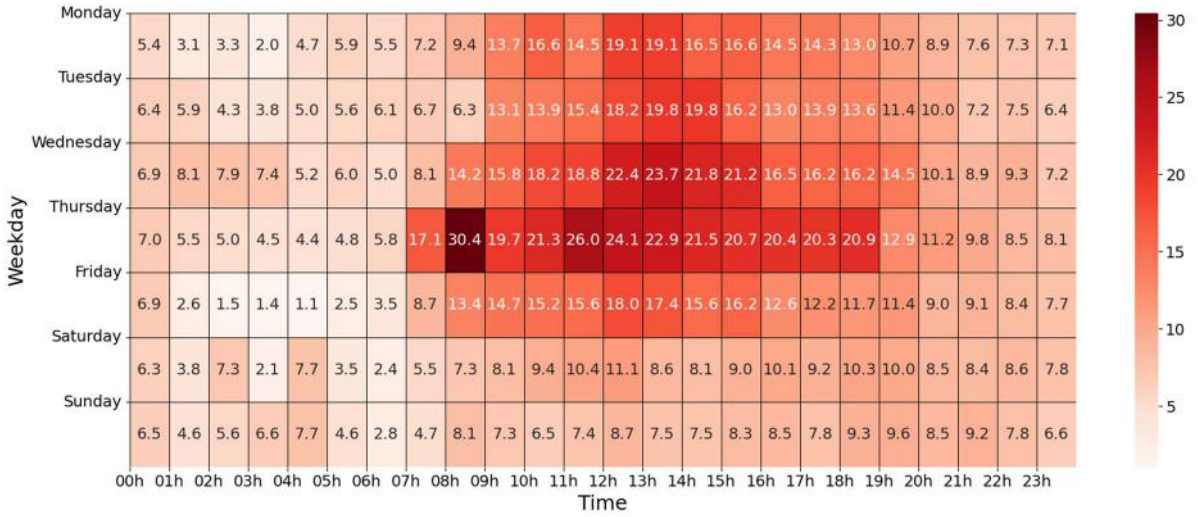
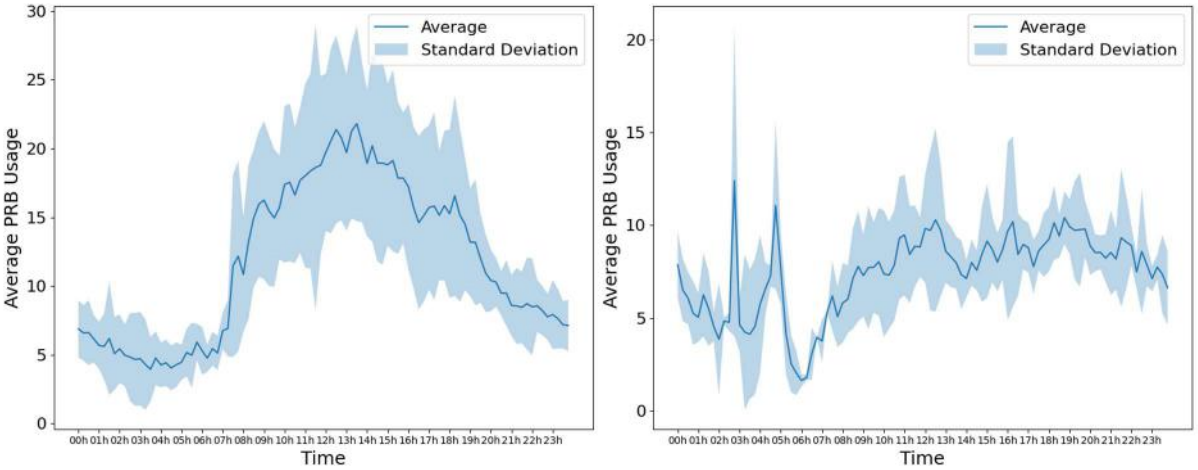


Figure 3.3 - Heatmap Pivot Table of Counter N_PRB_Used_AVG.

As observed in Figure 3.2 and 3.3, the usage patterns differ from the weekdays and the weekends, where weekdays consistently have more traffic than weekends. Consequently, weekdays and weekends were separated for each sector and two sets of statistics and plots were computed for each. Figure 3.4 shows an example of the average and standard deviation plot of the counter N_PRB_Used_AVG of a sector during the week (a) and weekend (b). The same thing was done with the other dataset features.



(a) Weekday. (b) Weekend.

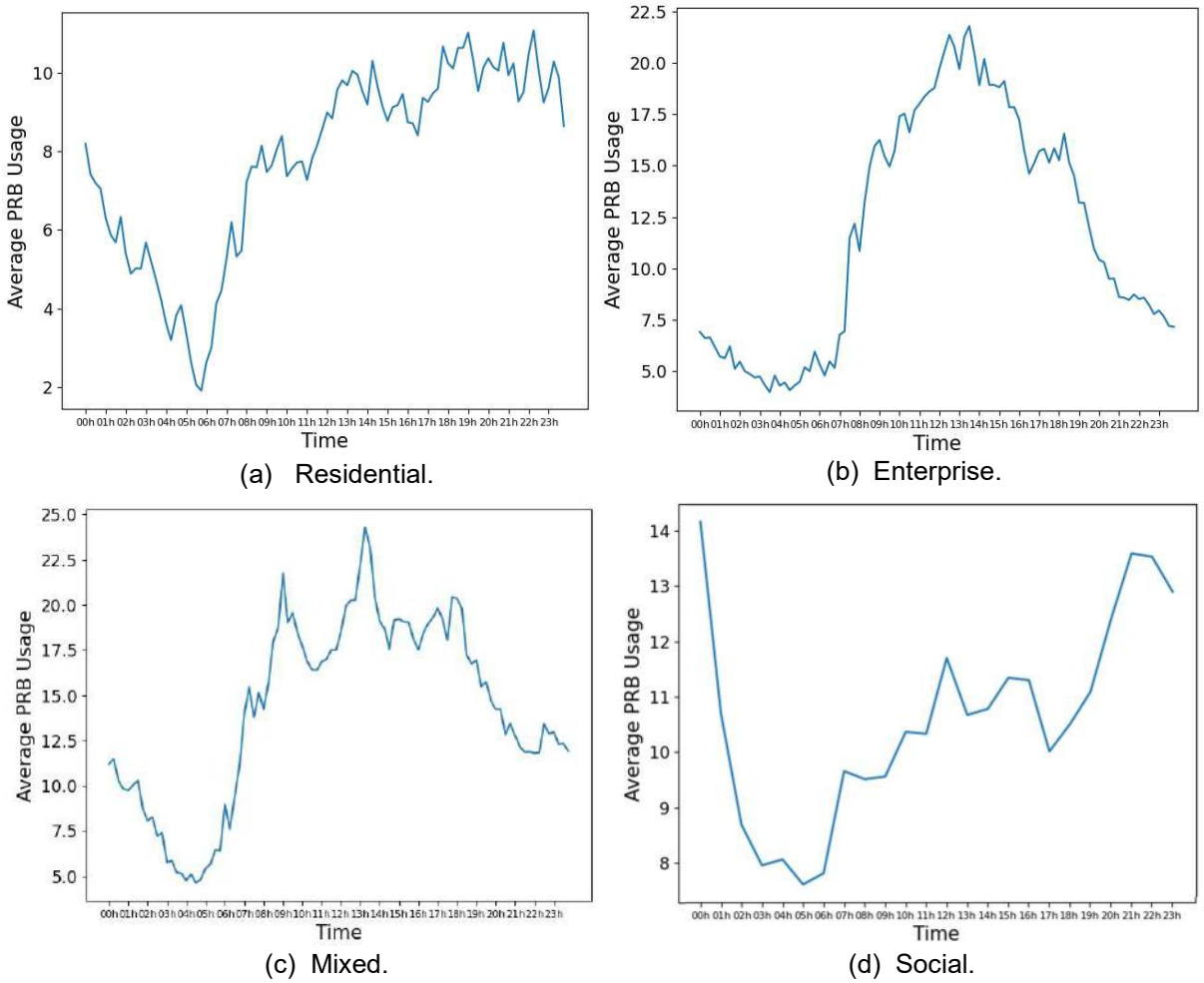
Figure 3.4 - Average and standard deviation plot of the counter N_PRB_Used_AVG, for a single sector.

As identified in Figure 3.2, the day-to-day usage follows similar patterns. Therefore, these statistics are helpful in identifying the general traffic patterns each sector has. When analyzing the different plots of the average values of the counter N_PRB_Used_AVG for different sectors and sites, four major traffic patterns surfaced:

- Residential: Lower traffic in the mornings, with a gradual increase during the day, with peaks early in the morning while leaving for work, during lunch time and after work, especially after

work hours and during the night when people are mostly active at home, streaming videos, using social media, and other online activities. An example is provided in Figure 3.4 (a).

- Enterprise: Traffic grows in the morning, typically at the beginning of work hours, peaking around lunchtime, and declining after work hours, staying low during the night. An example is provided in Figure 3.4 (b).
- Mixed: A combination of residential and enterprise patterns with daytime rise, post-work decline, but with sustained significant usage at night. An example is provided in Figure 3.4 (c).
- Social: Irregular traffic patterns with spikes during specific hours related to social activities, such as night-time peaks in entertainment zones, for example in our dataset a site with multiple beach bars, and daytime peaks in recreational and leisure areas. An example is provided in Figure 3.4 (d).



(a) Residential. (b) Enterprise.

(c) Mixed. (d) Social.

Figure 3.5 - Traffic Patterns Across the Dataset.

3.3.2 Time Series Analysis

A time series data analysis was conducted by examining its underlying components: trend, seasonality, and residuals along with investigating the autocorrelation properties of the series. Understanding these elements is crucial for effective future forecasting.

To decompose the time series into trend, seasonality and residuals the STL function from *statsmodels* detailed in [CCMT90] is used.

STL is a filtering procedure for decomposing a time series that uses locally estimated scatterplot smoothing (LOESS), which is a non-parametric regression technique used to smooth data that fits simple regression models (typically linear or quadratic) to localized subsets of the data, rather than to the entire dataset.

STL decomposes a time series into three key components:

- Seasonal Component: Captures repeating short-term patterns (such as daily, weekly, or yearly cycles). The seasonal component can evolve over time, making it suitable for real-world data with changing patterns.
- Trend Component: Represents the long-term direction of the series. By applying LOESS smoothing, STL captures slow-moving trends, allowing for smooth but potentially non-linear variations.
- Residual Component: The remaining "noise" or irregularities after removing the seasonal and trend components. This represents random fluctuations or anomalies in the data.

After decomposition, the time series can be represented as:

$$Y_v = T_v + S_v + R_v \quad , \tag{3.1}$$

where:

- v: current time step,
- *Y*: original time series,
- *T*: trend component,
- *S*: seasonal component,
- *R*: residual component.

The STL algorithm is an iterative process that obtains and slowly improves the three components. First, an initial trend estimate is removed from the time series, leaving behind a rough detrended version of the data. Then, LOESS is used on cycle-subseries (data points corresponding to the same period within each cycle, for example, each month in monthly data with yearly seasonality) to obtain the seasonal component of that cycle. To obtain the trend component, the previously calculated seasonal component is removed from the original series and the resulting series is smoothed using LOESS to obtain the trend. The STL algorithm alternates between estimating the seasonal and trend components until the estimates converge. After convergence, the residual component is calculated as the difference between the observed time series and the sum of the seasonal and trend components.

Figure 3.6 shows the decomposition of one week of PRB usage of a sector.

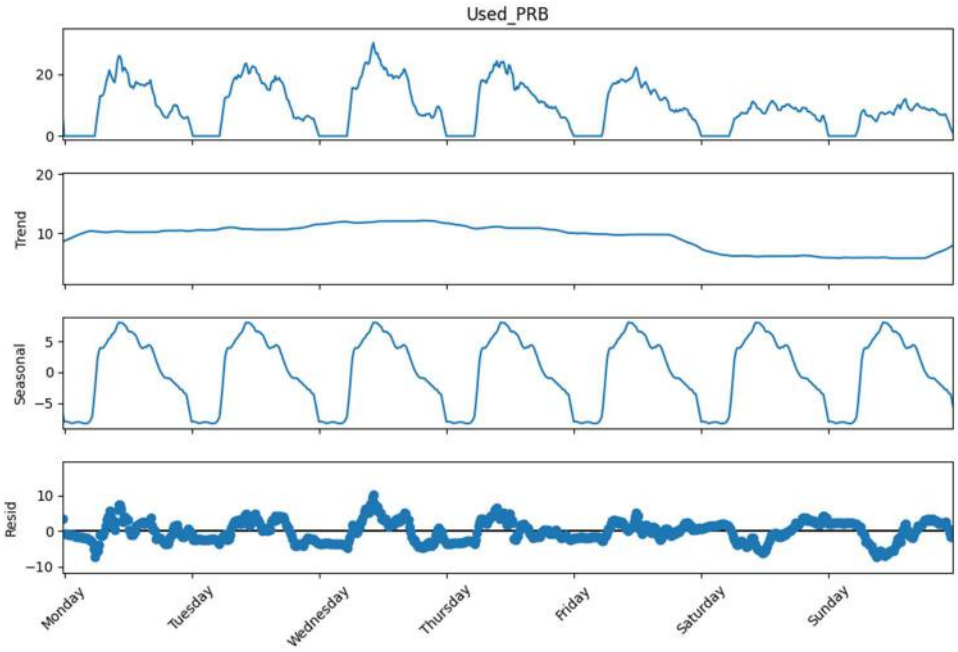


Figure 3.6 - Time Series Decomposition of the counter N_PRB_Used_AVG over a 7-Day Period.

As previously observed, network usage exhibits a regular pattern with noticeable peaks and troughs. The trend component captures the long-term movement in PRB usage. From the plot, it is evident that the trend is relatively stable during the week with a slight decrease in the weekend. This stability indicates that there are no significant changes in the underlying usage pattern, implying a consistent demand in the network over the observed period.

The seasonal component illustrates the periodic fluctuations that repeat every week. It is clearly observed that the data is highly seasonal. The residual component represents the random noise or irregular fluctuations after removing the trend and seasonal effects.

The residuals appear to be relatively large which indicates that there is seemingly unexplained variance in the data, suggesting the influence of unpredictable factors or noise that the trend and seasonal components do not capture. This likely happens due to the datasets having a large time granularity. Given this granularity, a substantial amount of traffic occurs within each time step. The increases and decreases of PRB usage within each interval are never the same, appearing almost random, as there will always be a slight variation in usage between each time step around the regular pattern. This inherent variability leads to significant residuals, indicating that even after accounting for trend and seasonality, a considerable portion of the fluctuations in PRB usage remains unexplained.

We also perform an autocorrelation analysis to uncover the underlying patterns and dependencies in our time series data. To conduct this analysis, the function *plot_acf()* from the *statsmodels* library on version 0.14.2 [Stat24] is used, which provides a visual representation of the autocorrelation function at various lags, with the autocorrelation for each lag plotted on the vertical axis and the lags plotted on the horizontal axis, shown in Figure 3.7.

The autocorrelation function calculates the correlation between a time series and its lagged version. The formula for the autocorrelation is:

$$\rho_a(h) = \frac{\sum_{t=1}^{N-h}(y_{t+h} - \bar{y})(y_t - \bar{y})}{\sum_{t=1}^{N}(y_t - \bar{y})^2} \quad , \tag{3.2}$$

where:

- y_t: time series at time *t*,
- $\bar{y}$: mean of the time series,
- N: number of observations in the time series,
- h: lag value.

The resulting values range from -1 to 1, where 1 indicates perfect positive correlation and -1 perfect negative correlation [BPDR10].

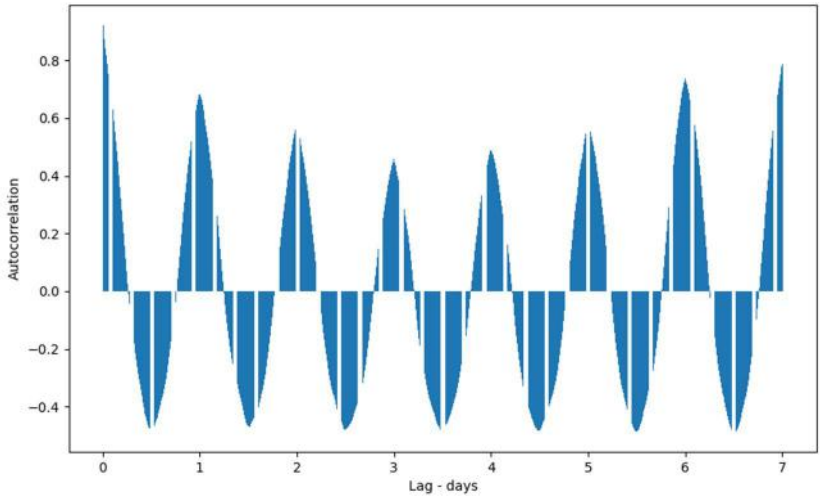


Figure 3.7 - Autocorrelation Plot of the counter PRB Usage Over a 7-Day Lag Period.

Analyzing this image, it can clearly be seen that autocorrelation peaks every 24-hour lag, with the highest autocorrelation value being with a 7-day lag. This indicates what was already concluded: the data is highly seasonal, with both a strong daily and weekly seasonality.

3.3.3 Feature Correlation and Selection

We now present the analysis of feature selection within our 5G base station traffic dataset. As the main goal of the thesis is finding energy saving opportunities in 5G networks, understanding the relationships between different features is crucial as it helps identify which variables are most strongly correlated with energy consumption, providing insights into the key drivers of energy consumption, as well as revealing redundant features, aiding in the selection of the most informative features for the predictive models which will help in finding the best time to apply the energy saving actions.

To confirm feature selection results, multiple methods were employed. The first feature selection method tried were filter methods. Both Pearson's and Spearman's correlation coefficients, detailed in (2.13) and (2.14), were computed for each sector. To help visualize the relationships between different features, correlation matrices were created. The correlation matrix displays the correlation coefficients between multiple features, indicating the strength and direction of their linear or monotonic relationships, with

values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). Figure 3.8 shows Pearson's correlation matrix of a sector of a site of the dataset of country A. Figure 3.9 shows the same thing but presenting Spearman's correlation instead. The observations in this example are representative of the correlation matrices for other sites and countries, all showing similar relationships.

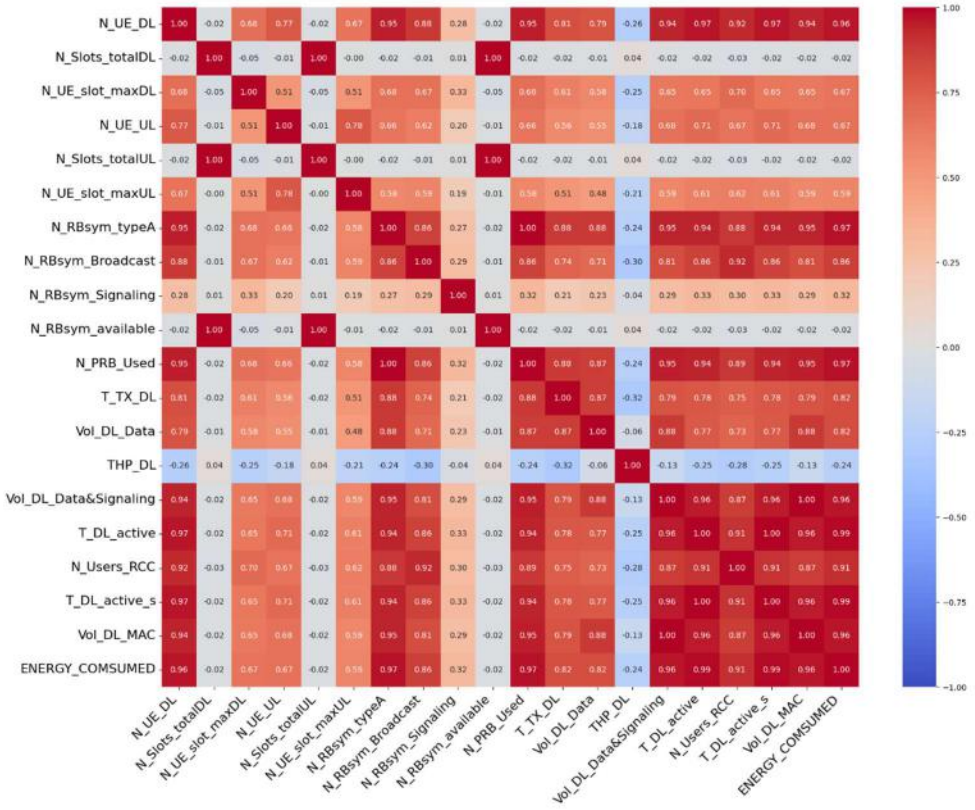


Figure 3.8 - Pearson's correlation matrix of a sector of the dataset of country A.

We can conclude that the features T_DL_active_s, Vol_DL_MAC, N_PRB_Used_AVG and N_UE_DL are the ones that are most correlated to the energy consumption of the site in both Pearson's and Spearman's correlation. Since no significant change happens from one correlation matrix to the other, it is safe to assume that the monotonic correlations captured in Figure 3.9 are the same linear correlations seen in Figure 3.8, meaning they are linear correlations. It can also be observed that the features with higher correlation are also highly correlated between one another, which can indicate that some of them might be redundant, which will be analyzed further. Other variables with high correlation, such as N_RBsym_typeA, T_TX_DL, Vol_DL_Data, etc., also have high correlation coefficients, but since they are directly correlated to the ones listed above, as they are featured in their calculation formulas, it can be assumed that these are redundant features. The same is observed in the correlation matrices of country B, where the features with highest correlation are the same, and variables such as Vol_DL_Cell, T_DL_Cell, etc., are considered to be redundant for the same reasons.

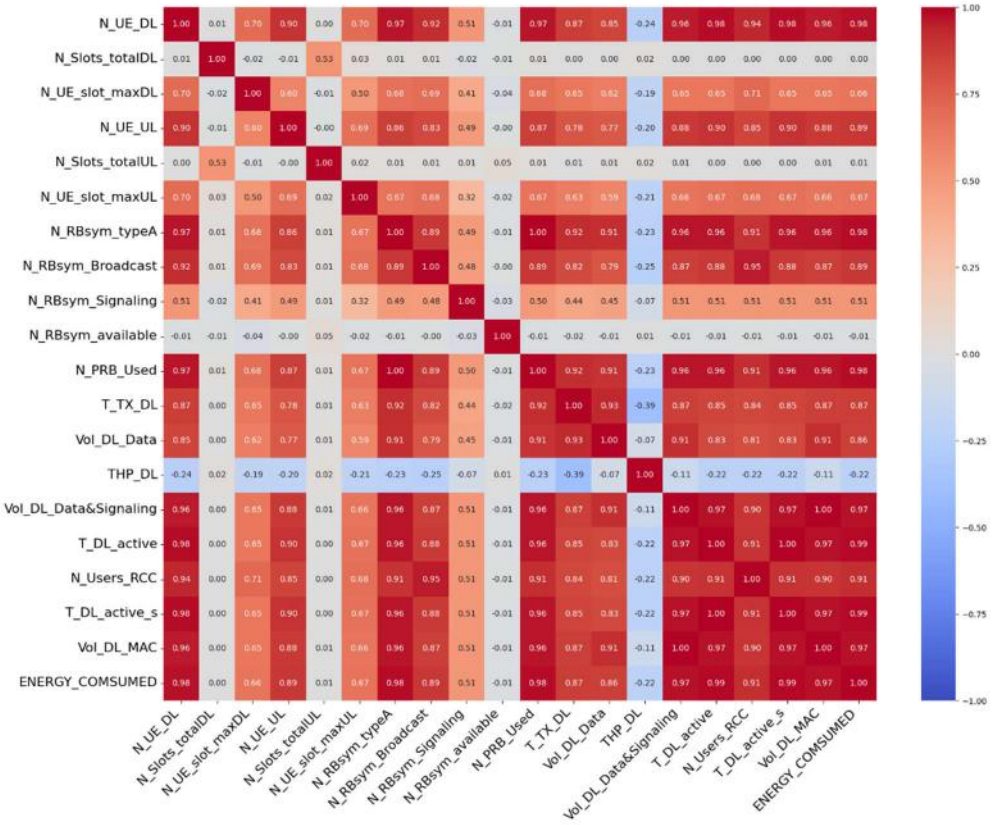


Figure 3.9 - Spearman's correlation matrix of a sector of the dataset of country A.

The next analysis was to compute the mutual information between each feature and energy consumption. Mutual information measures the dependence between variables, showing how much knowing one variable reduces the uncertainty about another. This helps us further solidify the relationships between features in our dataset. This analysis was performed for all sectors, and an interval of results were obtained, showing the maximum and minimum values obtained over all sectors followed by the average value. To calculate mutual information between features, *mutial_info_regression()* was used from the *scikit-learn* [Scik24] python library on version 1.5.2. For country A, the top 5 features with highest mutual information scores in relation to energy consumption are shown in Table 3.5:

Table 3.5 - Top 5 Country A features by mutual information.

Features	Max.	Avg.	Min.
T_DL_active_s	2.03	1.50	0.50
N_PRB_Used_AVG	2.05	1.31	0.42
N_RBsym_typeA	2.04	1.3	0.38
N_UE_DL	1.71	1.23	0.39
Vol_DL_MAC	1.60	1.17	0.39

For country B, similar information can be found in Table 3.6:

Table 3.6 - Top 5 Country B features by mutual information.

Features	Max.	Avg.	Min.
N_PRB_Used_AVG	3.03	2.00	0.75
T_DL_Active_s	2.29	1.52	1.07
T_DL_Cell	2.29	1.51	1.07
N_UE_DL	2.23	1.48	1.07
T_DL_Rmv_LastSlot	2.32	1.25	0.55

Similar importance results are obtained when comparing with correlation, with the clear presence of some redundant features in the top 5, which makes sense as these share similar information.

Ensemble feature selection methods, random forest and XGBoost, were used to identify the most influential features. The *RandomForestRegressor()* from the *scikit-learn* library was employed to obtain Gini importance scores. Since this implementation does not return permutation importance, the *PermutationImportance()* method from *scikit-learn* was utilized to assess the permutation importance. For XGBoost, the *XGBRegressor()* from the *scikit-learn* library was used, where the feature importance returned is the weight.

Table 3.7 shows the top 5 features for each model used for sites in country A, where an interval of values is presented, showing the lowest, highest followed by average values obtained from all sites. Table 3.8 shows the same for country B.

Table 3.7 - Top 5 feature selection scores for Country A.

Features	Random Forest Gini Importance			Random Forest Permutation Importance			XGBoost		
	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.
T_DL_active_s	0.96	0.77	0.03	1.47	1.06	0.13	0.93	0.72	0.01
N_PRB_Used_AVG	0.35	0.06	0.01	0.65	0.05	0.01	0.71	0.12	0.01
N_RBsym_typeA	0.46	0.05	0.01	0.26	0.03	0	0.53	0.07	0.01
DATETIME	0.23	0.03	0						
N_UE_DL	0.15	0.03	0	0.44	0.02	0	0.14	0.02	0
Vol_DL_MAC				0.17	0.01	0			
N_RBsym_Broadcast							0.47	0.03	0

Table 3.8 - Top 5 feature selection scores for Country B.

Features	Random Forest Gini Importance			Random Forest Permutation Importance			XGBoost		
	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.
N_PRB_Used_AVG	1.00	0.76	0.04	1.90	1.29	0.03	0.94	0.65	0.01
N_UE_DL	0.40	0.08	0	0.19	0.03	0	0.79	0.19	0
N_UE_UL	0.32	0.06	0	0.53	0.06	0	0.24	0.03	0
T_DL_Active_s	0.24	0.04	0	0.07	0.01	0			
T_DL_Cell	0.24	0.04	0	0.06	0.01	0	0.66	0.10	0.02
Vol_DL_MAC							0.04	0.01	0

Lastly, SFS, a wrapper method, is employed. We utilized Random Forest in conjunction with SFS method from *scikit-learn* to obtain a list of the top 5 features in our feature set. This method was executed for all sectors and the final feature set corresponds to the features that were chosen most times throughout the sectors. Both FFS and BFE was done, with the resulting selected features being the following, for country A:

- FFS:
    - DATETIME,
    - N_PRB_Used_AVG,
    - T_DL_Active_s,
    - N_RBsym_typeA,
    - N_RBsym_Signaling.
- BFE:
    - DATETIME,
    - N_PRB_Used_AVG,
    - T_DL_Active_s,
    - N_RBsym_typeA,
    - N_Users_RCC.

For country B the following features were selected.

- FFS:
    - N_PRB_Used_AVG,
    - N_RCC_Users_Avg,
    - T_DL_Active_s,
    - N_UE_DL,
    - T_DL_Rmv_LastSlot.

- BFE:
    - N_PRB_Used_AVG,
    - N_RCC_Users_Avg,
    - T_DL_Active_s,
    - N_UE_DL,
    - N_RCC_Users_Avg_DL.

From this analysis, it can safely be concluded that the most important features in the dataset regarding energy consumption are activity time, T_DL_active_s, and the number of used PRBs, N_PRB_Used_AVG, as the increased usage of a cell directly correlates with higher energy consumption. Other important variables are the number of active users, N_UE_DL, the volume of DL data, Vol_DL_MAC, and the RB symbols associated with data transmission, N_RBsym_typeA. These variables are highly correlated to each other, as can be seen in Figure 3.8, which again makes sense, as the more users there are in a cell, the more data is downloaded, the more PRBs are used and the higher the activity time in the cell. This can also be observed when using SFS, which is strong in eliminating redundant features, as some features that consistently rank with high feature importance scores are not selected, meaning they are mostly redundant and do not add predictive information.

Building upon these insights, it is concluded that the feature used as the main focus for our algorithm is the number of used PRBs, N_PRB_Used_AVG. The average PRB usage was chosen over the activity time because, while activity time simply measures the duration of cell activity, the PRB usage reflects the actual intensity of that activity, showing not just whether the cell is active but how much of its resources are being consumed. Therefore, N_PRB_Used_AVG will be employed both as a prediction variable and as a clustering variable in our analysis. By centering our algorithm around this key feature, we aim to effectively identify optimal times for implementing energy-saving actions that will have the most impact on consumption, ultimately enhancing the efficiency of the network.

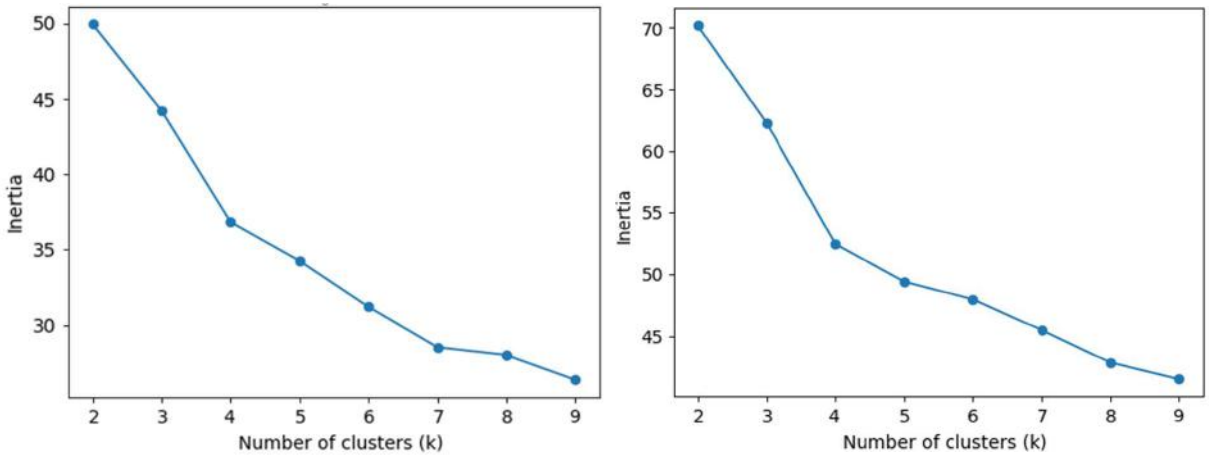
3.4 Traffic Patterns Clustering

Balancing the trade-off between model simplicity and performance is crucial. A "one size fits all" approach, where a single model is trained to handle all sectors, may prove inadequate. Traffic patterns can vary significantly across sectors, making it difficult for a single model to generalize well, resulting in poor performance. On the other hand, training a unique model for each sector, while potentially offering better performance, introduces complexity. This approach may be more computationally expensive, difficult to manage, and challenging to implement on a scale. The training and maintenance of such a large number of models can be resource-intensive and time-consuming, especially if network conditions and patterns change frequently (summer and winter patterns). By grouping sectors with similar weekly and weekend traffic patterns, the number of models needed are reduced, making the solution more efficient and easier to implement. This approach aims to maintain performance without overcomplicating the algorithm or significantly increasing computational costs.

The traffic patterns obtained in Section 3.3.1 were grouped into clusters of similar patterns using the *KMeans()* algorithm of the *scikit-learn* library.

Visually, four distinct traffic patterns can be identified. However, as there may be a more optimal number of clusters, it was decided to not limit the clustering algorithm to this observation. To determine the best grouping, the elbow method [TWTH01] using inertia was employed to identify the optimal split, ensuring the clustering is driven by the data rather than a fixed assumption of four patterns.

The elbow method is a technique used to determine the optimal number of clusters in a dataset. It works by running a clustering algorithm multiple times with different numbers of clusters and calculating the inertia, that measures how tightly data points in a cluster are grouped, for each run. As the number of clusters increases, the inertia decreases because more clusters mean less distance between points and their assigned cluster centers. However, after a certain point, adding more clusters results in only marginal improvements. The optimal number of clusters is identified at the "elbow point," where inertia starts to decrease at a slower rate, indicating diminishing returns. This point is where the data is best grouped without overcomplicating the model.



(a) Weekday patterns. (b) Weekend patterns.

Figure 3.10 - Elbow method for weekday and weekend patterns.

This can be seen in Figure 3.10, for both the weekday and weekend patterns, where a noticeable bend occurs at 4 clusters, indicating the optimal number of clusters. Another bend can also be seen in the weekday patterns at 6 clusters. Consequently, this division of the series into 6 groups will also be trained and compared, along with other grouping numbers, to evaluate the performance and determine the most suitable clustering solution.

Figure 3.11 depicts the computed weekday clusters and their centroids with 4 clusters, where the black line is the cluster centroid, and the colored lines are the patterns of the sectors that belong to the cluster.

The sectors from each cluster are grouped together into a combined dataset that will be used for training the prediction models.

The *KMeans()* algorithm is also used to assign new traffic patterns into the previously obtained clusters. This ensures that the most appropriate prediction model is selected, therefore optimizing the accuracy and effectiveness of the results.

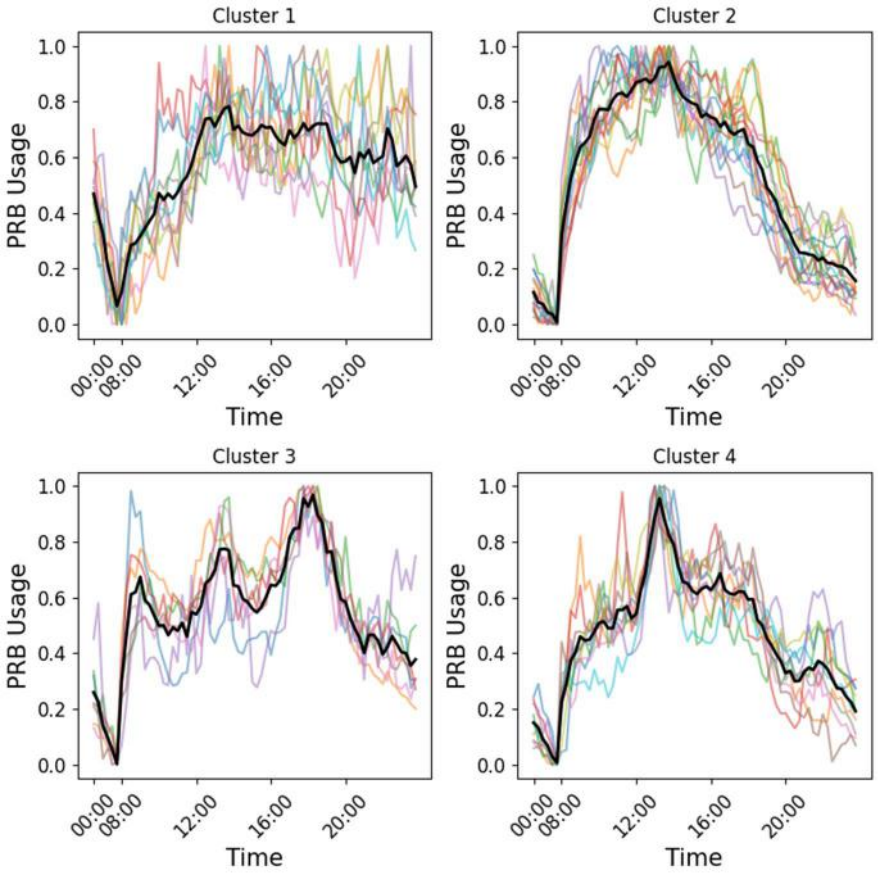


Figure 3.11 – Weekday pattern clustering into 4 clusters.

3.5 Traffic Prediction

In order to improve the capability of our energy saving methods, traffic prediction is utilized so pre-emptive measures can be taken when appropriate. LSTM networks were chosen for this purpose due to their proven effectiveness in time series prediction. Our goal is to use our dataset as multiple time series to predict the future behavior of the network, helping us take steps to save energy.

3.5.1 Data preparation

To prepare the data for input into the LSTM model, it is first organized and structured to align with the model's learning requirements. This involves formatting, normalizing, and segmenting the data into sequences that reflect the temporal patterns necessary for accurate prediction. Each data preparation step is done individually for each sector. To begin with, the original dataset is divided based on the number of traffic pattern clusters chosen, where each dataset contains the information of the sectors of the corresponding cluster.

First, since, as seen before, each sector turns off during the night, these hours are removed from the dataset. This is done so that the model focuses on the relevant patterns and predictions and not on these hours where the outcome is already known. The datasets were also divided into weekdays and weekends, in order to prevent confusing the predictions of the model, as these show different traffic patterns and there is not much data to train on.

Each dataset was divided into three: train, validation and test datasets. The split was designed so that the test dataset consists of the last 7 days of the data, which begins on Saturday and ends on Friday and is approximately 23% of the entire dataset. The remaining data was then divided between the training and validation sets in a 90/10 ratio. The train and validation datasets act as historical data, where the train dataset is used to train the model, and the validation dataset is utilized primarily to prevent overfitting. Specifically, the validation dataset serves as an early stopping metric that halts the training process once performance on the validation set stops improving, ensuring the model does not overfit to the training data. The test dataset will represent the future behavior of the network and will be used to compare and evaluate the predictions of the model against the real values. This division kept the values in the original order as our goal is to capture the temporal dependencies in the data.

The data was then scaled to prepare it for training the LSTM model. This step helps in the learning process by ensuring that the values from different sectors, which may vary significantly in range, are brought within the same scale so that the model focuses on learning the temporal patterns in the data, rather than being influenced by the varying value ranges of different sectors. To do this the *StandardScaler()* function from the *scikit-learn* library is utilized on each feature, transforming the data so that its values are centered around 0 with standard deviation of 1, effectively standardizing it. A separate scaler was fitted to the training dataset of each sector to simulate a real environment and ensure more accurate test results, as only historical data would be available during training. The validation and test sets were then standardized using the corresponding scaler fitted to the training data. Data Standardization was chosen instead of normalization, since a maximum value for the data cannot be defined, as values coming from the test dataset can be larger than those seen during training. Additionally, the scaler for each sector was saved so that it could later be reverted back to its original range when needed.

In order to feed the data to the LSTM for training, it has to be reshaped. LSTMs utilize supervised learning as their learning method, meaning that the time series prediction problem must be re-framed as a supervised learning problem. To do this, one needs to come up with inputs and outputs for the model. In our case, our objective is to be able to predict the next time step of a chosen feature, given the previous ones. We assign X, the input, as the previous time steps one wants the model to observe, where the parameter *seq_lenght* specifies how many time steps the model should look back, and Y, the output, as the current time step that the model is trying to predict. In our case, X contains more than one feature sequence, meaning that multiple features from the dataset are used to create different sequences that the model will look back to aid in prediction. This is done for the train, validation and test datasets separately.

The last data preparation step was feature engineering time features. In order to give the model a better understanding of the time of the day and week of the current step, so that it better recognizes temporal patterns, four time features were engineered using (3.3) and (3.4).

$$p_{sine} = \sin\left(s \times \frac{2\pi}{t}\right) \ , \quad (3.3),$$

$$p_{cosine} = \cos\left(s \times \frac{2\pi}{t}\right) \ , \quad (3.4),$$

where:

- *s*: timestamp in seconds of the current date taken from the date-time feature,
- *t*: number of seconds present in a day or week.

These return values between -1 and 1, where each timestamp is associated with the current progress of the day or week, where 1 represents the beginning of the day (midnight) or week (Monday midnight) and -1 represents the halfway point of these two.

3.5.2 Model Development

Multiple single-step forecasting LSTM models were tested, varying architectures, hyperparameters, and features to identify the best-performing configuration for predicting network PRB usage. The aim was to select a model that could accurately capture usage patterns, enabling the identification of low-usage periods to support energy-saving measures in network management.

To have a baseline comparison of the performance of the model, three baseline models were created. These are simple non-sophisticated and non-predictive models that serve as a reference point for the LSTM model as a minimum performance threshold:

- Naive Model: A simple naive prediction model was developed. It uses the last observed value as the prediction for the next value and can be represented with (3.5).

  $$\hat{y}_{t+1} = y_t \ , \quad (3.5)$$

  where:

  - $\hat{y}_{t+1}$: predicted value at time *t*+1,
  - y_t: observed value at time *t*.
- Seasonal Naive Model: Since our time series have clear seasonality, a seasonal version of the previous model was developed. The seasonal naive model takes the same idea from the regular naive model, but instead of having the last observed values as the prediction, it uses the series seasonal patterns, taking the value of the same season previous cycle as the predicted value. It can be represented with (3.6).

  $$\hat{y}_{t+1} = y_{t-s} \ , \quad (3.6)$$

  where:

  - y_{t-s}: observed value from the previous season corresponding period,
  - *s*: seasonality period.
- Moving Average Model: This model predicts the future value based on the average of a time

window of the previous 4 time-steps. It can be represented with (3.7).

$$\hat{y}_{t+1} = \frac{1}{n}\sum_{i=0}^{n-1} y_{t-i} \quad , \tag{3.7}$$

where:

- y_{t-i}: historical observations,
- *t*: current time step,
- *n*: window size.

Multiple LSTMs models were created and trained using the Python *TensorFlow* [Tens24] and *Keras* [Kera24] libraries on versions 2.17.0 and 3.5.0, respectively. In a first phase three different models were created:

1) Univariate model with only the target features historical data as input.
2) Multivariate model with the target features historical data and engineered features as input.
3) Multivariate model with the target features historical data, engineered features and other selected features with high correlation with the target feature, such as cell activity time, number of connected users and data volume, as input.

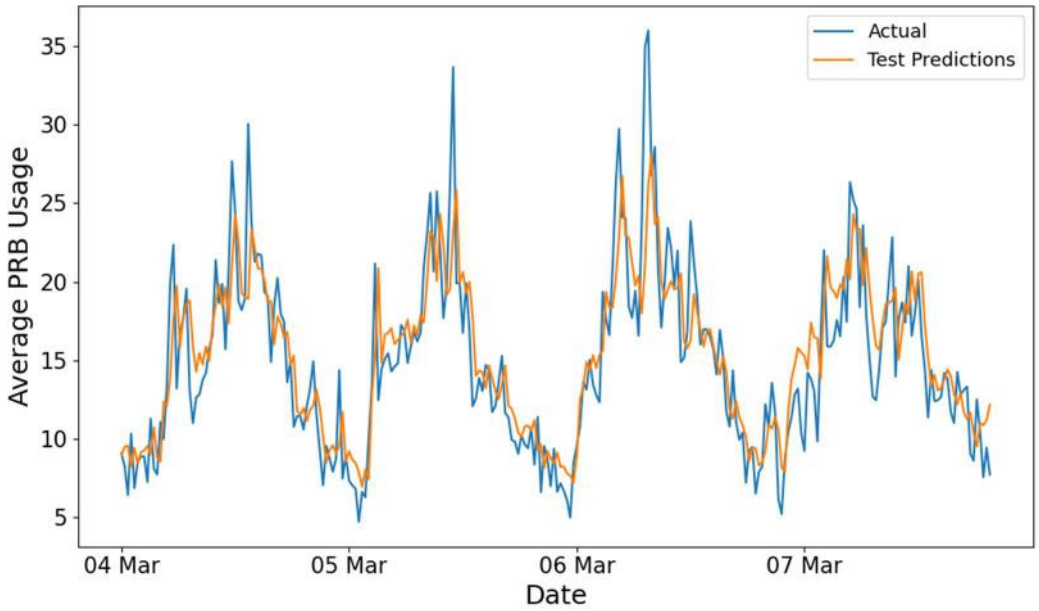


Figure 3.12 - Test predictions of model 3).

Upon initial inspection, the predictions of the model appear to be highly accurate, closely following the actual values, as seen in Figure 3.12, where both a portion of the test dataset along with the corresponding test predictions are shown. However, when zooming in on predictions it can be observed that the predictions exhibit a consistent delay. This phenomenon is most noticeable during instances of steep changes from one time step to the next, as seen in Figure 3.13. The predictions of the model tend to align closely with the values from the previous time step, suggesting that the LSTM is functioning similarly to a naive forecasting model.

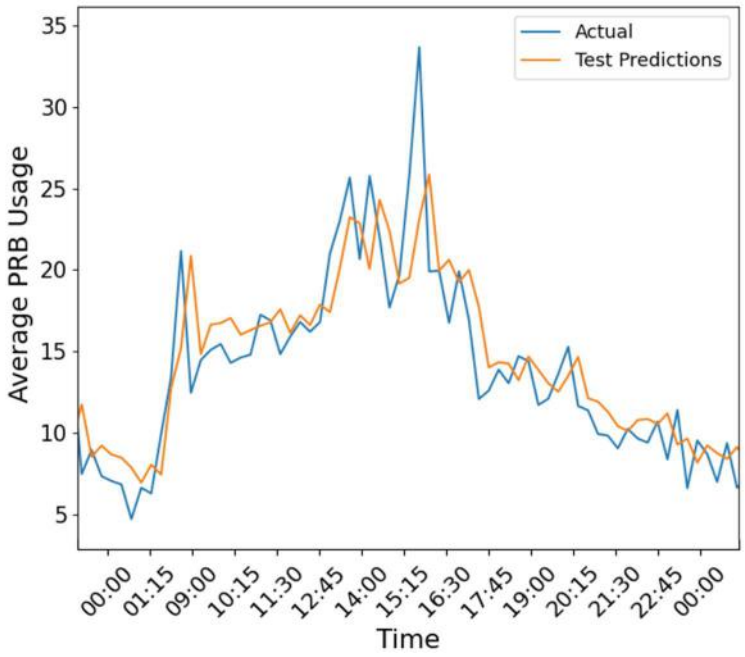


Figure 3.13 - Close up of the predictions of model 3).

This behavior indicates that when attempting to predict the next time step, the LSTM struggles to identify any directional trend and defaults to a strategy of predicting values that are near the previous observation, which minimizes prediction error. Despite the high autocorrelation and significant seasonality present in our series, as previously concluded, the large residuals present in our data, which are close to a random time series, hinder the ability of the LSTM to discern meaningful patterns, leading it to default to a naive approach that minimizes the error.

It can be concluded that these models do not effectively learn the patterns present in the data, leading to predictions that lack meaningful insights. As a result, the effectiveness of the model as a predictive tool is compromised, as its performance is only marginally better than that of a naive forecasting model.

A different approach had to be taken to obtain a more effective prediction model. In a second phase, additional models were developed to attempt to mitigate the issues:

4) All the first phase multivariate models were redone, now removing the historical data of the target feature as an input.
5) Multivariate model where the target is a smoothed version of the original one, computed using moving average, also using historical data of the original version of target feature, engineered features and selected dataset features such as cell activity time and number of connected users, as input.

These models attempt to resolve the previous issue in different ways. Since the previous predictions of the model followed the latest time step, which was given to the model as historical data, model 4) attempts to prevent it by removing the historical data of the target feature. Model 5) tries to solve the problem of the large residual present in the data by smoothing it with the goal of providing the LSTM with a cleaner version of the dataset, devoid of the high residuals, allowing it to focus on the evolution

of a smoother, more general traffic pattern, and not the specific fluctuations that are always changing from day to day.

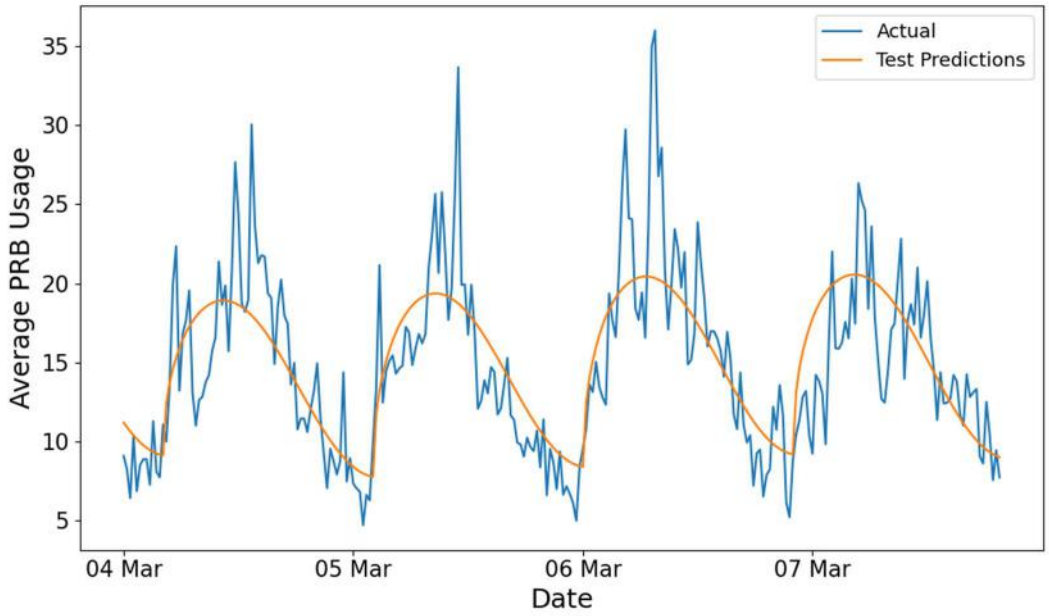


Figure 3.14 - Test predictions of model 4).

Looking at model 4) predictions in Figure 3.14, it is clear to see that, without the target feature historical data present, the model has a hard time making correct predictions, further confirming the hypothesis that these models cannot learn meaningful insights from the data.

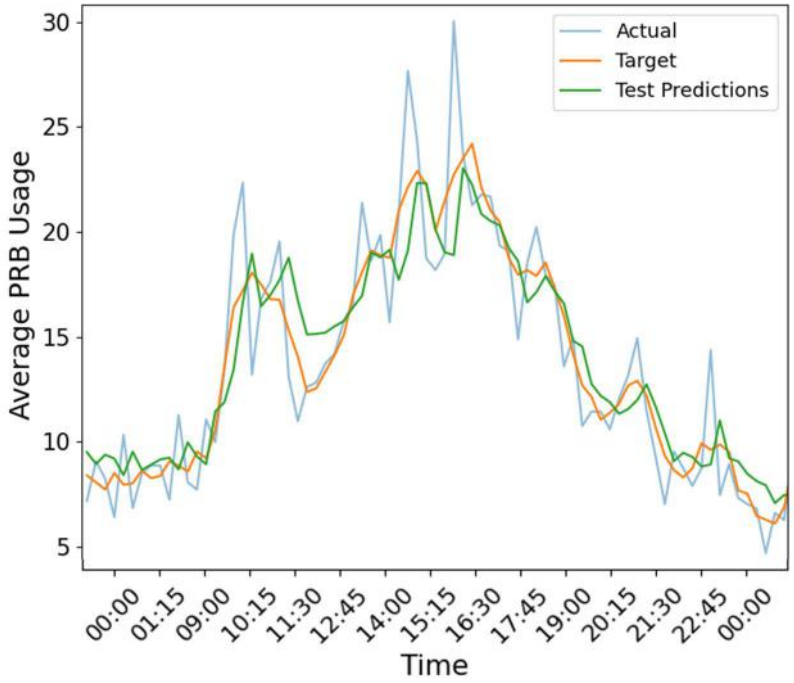


Figure 3.15 - Test predictions of model 6).

Shifting focus to model 5) shown in Figure 3.15, the predictions show improved alignment with the target feature values, eliminating the consistent delay observed in the first phase models. However, the real

performance of the model is only described by the comparison between the predictions and the real traffic values, not the target feature (moving average of the real traffic values). In this case, the same delay seen from previous models can be seen, but significantly lower.

While the use of a moving average reduces the noise, greatly improving the ability of the ability of the model to learn the general patterns of the data (since predictions follow very close to the target), it still introduces some delay, as increases and decreases in values occur more gradually due to the smoothing effect.

Although this approach solves the issue of the predictions simply following the previous time step, the delayed response remains problematic when aiming to predict real-time target values. Despite this, the overall pattern captured by the model remains valuable. It provides a broader understanding of traffic trends, which is critical for the actions of our algorithm. For our application, what matters most is not the small fluctuations in predictions but the general trend they follow. Large errors, especially during peaks or dips, could have a more substantial impact on energy-saving measures. Smoother, more generalized predictions align better with the overall traffic pattern, reducing the likelihood of significant errors that could affect system performance. This ended up being the final model used.

In order to create the best model, hyperparameter tuning was done using the *GridSearchCV()* function of *scikit-learn*. This function systematically works through multiple combinations of parameter, cross-validating as it goes to determine which tune gives the best performance. The grid search was performed using Adam as the optimization algorithm and the cross validation was done using the *TimeSeriesSplit()* function of the *scikit-learn* library to account for the temporal structure of the data, ensuring that the model is validated on unseen time periods without shuffling the data, preserving the important chronological order. The following parameter grid was defined using 3-fold cross validation:

- *Seq_lenght*: 1 time step, half a day, one day, one week.
- Nº of units: 1, 10, 50, 100, 200, 500.
- Nº of layers: 1, 2, 3.
- Batch size: 32, 64, 128.
- Activation functions: ReLu, Tanh, Sigmoid.

The specific grid of *seq_lenght* values are chosen so that the autocorrelation peaks (daily and weekly) are captured by the model. These values are converted to time steps, depending on the time granularity of the dataset.

After performing the grid search, the following optimal architecture was achieved:

- *Seq_lenght*: One day.
- Nº of units: 200.
- Nº of layers: 1.
- Batch size: 32.
- Activation function: Sigmoid.

To prevent overfitting and achieve the best model performance, the *EarlyStopping()* function from the *TensorFlow* library was utilized. Early stopping monitors the validation loss during training and stops the

training process once it stops improving, preventing the model from overfitting. Early stopping was implemented with a patience of 75 epochs, meaning that if the validation loss did not improve for 75 consecutive epochs, the training stopped, and the weights of the model were restored to the best ones recorded during training, ensuring optimal performance without overfitting. In this case, the models trained on average for two to three hundred epochs.

3.6 Confidence Interval of Predictions

Predictions are never perfect, and errors can lead to incorrect decisions, like turning off a network sector when it should not be, which could disrupt the network. As seen already, the received dataset contains large residuals since, beyond the general trend and seasonality of the data, the hourly fluctuations are almost random from day to day. This makes confidence intervals crucial to account for the uncertainty in predictions.

Confidence intervals add flexibility to our algorithms by allowing the operator to adjust how strong the predictions are. A larger confidence level makes the model more cautious, ensuring fewer false actions but potentially missing some opportunities. Conversely, a smaller confidence interval makes the model more aggressive, reacting more often but with a higher risk of unnecessary interventions. This configurability lets operators fine-tune the model to balance the trade-off between sensitivity and caution, adjusting it to the specific needs of the operator.

A confidence interval of a prediction represents the range of values within which the prediction is likely to fall, essentially reflecting the uncertainty in the prediction. It indicates how much error one can expect, depending on the level of confidence one wants to achieve. By analyzing the past prediction errors one can assess an appropriate confidence interval based on their frequency or probability.

To do this, the method presented in [HRAG18] is used. It proposes calculating the confidence interval by combining the standard deviation of the prediction errors with the Z-score corresponding to the desired confidence level if the errors follow a normal distribution.

The prediction errors for each sector are tested for normality, which shows that they do follow a normal distribution. Consequently, the confidence interval can be calculated by using the following formula [Simu08]:

$$I_{CI} = X_t \pm Z \times \frac{s}{\sqrt{N}} \quad , \tag{3.8}$$

where:

- X_t: current prediction,
- *Z*: Z-score,
- *s*: sample standard deviation of the prediction errors,
- *N*: prediction errors sample size.

To obtain the Z-score, the *norm.ppf()* function from the *scipy.stats* module is used, which returns the Z-score associated with the chosen confidence interval.

Since the model does not have access to future predictions, the initial standard deviation is calculated using the prediction errors from the validation set. As predictions from the test set are generated, the associated errors are incorporated, and the standard deviation is subsequently recalculated. This iterative process allows for a more accurate estimation of uncertainty over time, reflecting the performance of the model as it encounters new data.

Figure 3.16 illustrates the 95% confidence interval, represented by the shaded gray area over multiple time steps, where each time step corresponds to 15 minutes, in this case. This region depicts the uncertainty of the predictions, encompassing 95% of the possible outcomes within this interval.

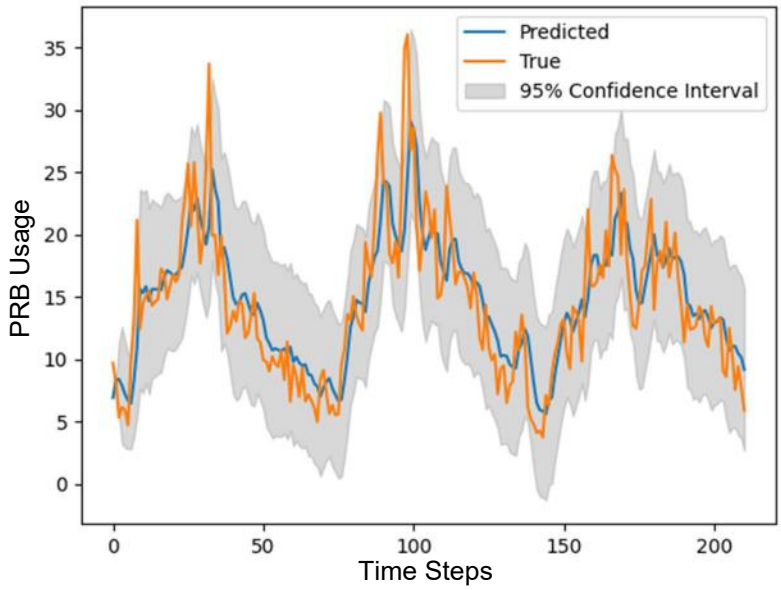


Figure 3.16 - Predictions 95% confidence interval.

3.7 Energy Saving Algorithm

This section presents a comprehensive explanation of the developed algorithm, connecting all prior discussions to demonstrate how it functions. The actions taken by the algorithm, along with its benefits, potential drawbacks and evaluation metrics are described in detail. The section is divided into two parts: the first focuses on the current algorithm, explaining its structure and functionality, while the second explores potential future implementations, introducing new ideas that align with the reasoning behind the developed approach.

3.7.1 Current Implementation

To achieve tangible energy savings, it is necessary to develop an algorithm that implements actions within the network and evaluates their effects on energy consumption and their impact on overall network performance.

First, the actions performed by the algorithm were defined. Given the large time granularities in our datasets (15 minutes to 1 hour), many potential energy-saving measures, such as turning off specific network components, would not be practical, as these actions require a much finer time resolution to be

effective, as previously discussed. After careful analysis, a feasible action was identified: Turning off the RRU. This action is already implemented by the operator in the provided datasets, where the RRUs are typically turned off during nighttime hours. Our algorithm will enhance this approach by dynamically deciding whether to keep the RRU powered on or off at each time step, allowing for more responsive energy management based on real-time traffic patterns.

The rationale behind the first action, turning off the RRU, is that the network already incorporates similar features, allowing us to build upon existing practices. This approach enables us to analyze the potential benefits and drawbacks of network energy consumption using the information provided by the datasets that contain historical data on RRU energy consumption during nighttime periods. With this, one can gain insights into the impact of this action on energy consumption.

These actions are triggered by an operator defined threshold of load capacity, defined by the proportion of Used PRBs relative to the available PRB, in percentage. The algorithm takes the predictions made by the previously developed LSTM and compares it to the operator defined threshold. If the predicted PRB usage falls below this threshold, the corresponding actions are performed. This threshold helps ensure that the energy-saving measures do not adversely impact the network, allowing it to operate as intended by the operator while efficiently managing energy consumption.

When an RRU is turned off, traffic data for the affected sector is effectively lost, which may impact future predictions, as the LSTM model will no longer have access to the previous expected usage data for that sector. To mitigate this, one approach is to use the LSTM's predictions to fill in the gaps during these periods of no data, providing continuity and improving prediction accuracy. In scenarios where this approach is insufficient, more advanced methods can be implemented to prevent significant deviation from real usage patterns. In our case, the algorithm utilizes actual sector usage data during these off periods, simulating a scenario where a more sophisticated prediction method is applied to closely approximate real traffic, ensuring minimal prediction drift from reality.

This action presents its own set of benefits and drawbacks that must be carefully analyzed. This analysis is essential for providing comprehensive feedback on both the energy savings achieved and the overall effectiveness of the algorithm.

The benefit of this action is the generated energy savings. Since our datasets already contained energy-saving features similar to the action of turning off the RRU, measuring the impact on energy consumption was straightforward. For each sector, the dataset includes a feature that records energy consumption at each time step during the period when the RRU is turned off, and this value remains constant. Therefore, when the first action of turning off the RRU is implemented, one can assume that energy consumption defaults to this recorded value, allowing for a clear assessment of the energy savings achieved through this measure.

Regarding the drawbacks, turning off the RRU means that the smaller number of users serviced by that sector must be reassigned to other frequencies or technologies. This relocation not only increases the load on those alternative resources but also results in higher energy consumption elsewhere in the

operator's network, which must be carefully considered when evaluating the overall energy savings and effectiveness of the proposed actions.

When evaluating the impact of turning off an RRU in a 5G network, it is crucial to account for the difference in energy consumption between the 5G technology and other networks, for example, 4G. We assume that when the RRUs are turned off, the traffic is routed to a 4G cell in the same site. In a typical 5G deployment, the advanced signal processing tasks, such as beamforming, massive MIMO, and managing higher data throughput, account for approximately three-fourths of the total energy consumed by the RRU. This estimate is based on the antenna specifications available for the sites under consideration. By contrast, this means 4G networks, where these complex processes are not required, RRU energy consumption is significantly lower, constituting around one-fourth of the total energy. This energy difference provides the basis for calculating the effects of turning off an RRU and reallocating users to other frequencies or technologies. If an RRU is deactivated, the assumption can be made that the energy consumption that would have to be served by a 4G BS would only be one-fourth of the total energy spent by 5G, meaning that even though there is a negative impact associated with the routing of traffic to other technologies, there is still a net gain in energy savings.

This added energy consumption estimation is considered to be a worst-case scenario. When an RRU is turned off, users must be reassigned to other network sites, which are typically already operational. The added energy consumption from managing this increased traffic is generally lower than that of a completely turned off RRU, as even with no traffic, RRU energy consumption is still considerable. This leads to a lower overall impact on neighboring sectors compared to the worst-case estimate. It is important to note that this calculation remains an approximation, providing a starting point for assessing the impact of turning off RRUs. Actual network performance and energy data will be needed to derive more accurate figures, but this approximation allows for a general understanding of the energy consumption dynamics between 5G and 4G and the trade-offs of reallocating users across the network.

To assess the performance of this algorithm, some metrics were developed:

- Hit ratio: The hit ratio measures the proportion of correct actions made by the model when the predictions are within the threshold defined for action. It can be defined as:

$$\eta_H = \frac{H}{N} \quad , \tag{3.9}$$

  where:

  - *H*: number of correct predictions that led to correct action,
  - *N*: total number of predictions that led to action.

- Miss ratio: The miss ratio measures the proportion of incorrect actions made by the model when the predictions are withing the threshold defined for action. It can be defined as:

$$\eta_M = \frac{M}{N} \quad , \tag{3.10}$$

  where *M* is the number of incorrect predictions that led to incorrect action.

- Accuracy: Accuracy measures the proportion of correct actions out of the total number actions that could have been taken. It provides a general assessment of how good the algorithm is at finding all possible action opportunities. It can be defined as:

$$A = \frac{TP}{TP+FN} \quad , \tag{3.11}$$

where:

- *TP*: true positive actions,
- FN: false negative actions.

which summed up amount to the total action opportunities possible.

3.7.2 Possible Improvements

The action of turning off the RRU has potential for reducing energy consumption but comes with the drawback of needing to reassign the users of the sector to other frequencies or technologies. This reassignment increases the load on alternative network resources, which may result in higher energy consumption elsewhere, limiting the overall energy savings.

Observations from the dataset notes that across each cluster, the energy consumption of RRUs in the 32/32 MIMO sectors was lower than that of the 64/64 MIMO configuration sectors. With this in mind, a different approach could be explored for future work that avoids this drawback, while using the same data. One possible action would be to deactivate specific antenna elements within sectors that use a 64/64 MIMO configuration. By reducing a 64/64 MIMO sector to a 32/32 configuration, the energy consumption can be lowered without completely shutting down the sector. This would allow the sector to remain functional, supporting network traffic, but at a reduced capacity, which is ideal for periods of lower demand.

The dataset includes sectors from the same cluster, which utilize similar equipment and are in close proximity, leading to closely aligned energy consumption values. Based on this information, energy consumption for the 64/64 MIMO sectors can be simulated by assuming they will behave similarly to the 32/32 MIMO sectors when some antenna elements are turned off. Using the energy consumption patterns observed in the 32/32 MIMO sectors, the energy usage for a 64/64 sector can be approximated under reduced antenna configuration. Furthermore, because the dataset reveals a high correlation between various features and energy consumption, the energy usage can be modeled for any sector under modified conditions using these correlations, enabling us to estimate energy savings accurately based on the prevailing traffic at each time step, the same way done when turning off the RRUs.

Implementing this action resolves the drawback of traffic rerouting and increased energy consumption elsewhere in the network. However, it introduces other challenges. Reducing the number of active antenna elements from 64/64 MIMO to 32/32 MIMO decreases the user capacity of the sector, which could affect how many users can be served at the same time. There may also be degradation in signal quality and coverage due to fewer active antennas. These factors should be carefully considered, as they could impact the overall user experience and network performance.

With the data available in the current datasets, the full effects of this action cannot be accurately measured or predicted. The lack of detailed traffic data and specific network insights makes it challenging to quantify how reducing MIMO configurations will affect user capacity, signal quality, and overall network performance. For these reasons, this action remains a conceptual proposal. Future work should focus on collecting more detailed data to explore the impact of this approach and to assess whether it is a viable energy-saving strategy while maintaining the quality of service.

Another improvement that can be achieved in the future is addressing the issue of missing sector usage data when an RRU is turned off, which is critical for maintaining accurate predictions. When an RRU is deactivated, predictions are needed to fill in the time steps where actual usage information is unavailable. Using predictions from the LSTM model developed in this work, or even incorporating predictions from alternative models, can help fill in these gaps. However, this approach may be limited, especially if the sector remains off for extended periods, as this results in substantial data loss that could significantly degrade predictive accuracy. To further mitigate this effect, future research could explore the integration of external information sources, such as usage data from other frequencies and technologies co-located at the same site. By leveraging data from 4G or other nearby frequencies, a general usage demand pattern can be inferred, providing an estimate for the missing 5G data. This additional layer of information would enable a more comprehensive view of the site's overall demand, supporting the LSTM’s predictions and helping to counteract the negative impact of data loss due to RRU deactivation.

Chapter 4

Analysis

This chapter presents an analysis of the performance of the algorithm under varying conditions, including different parameter settings and traffic scenarios. The results are compared with other models and baselines to assess its overall effectiveness. This analysis is conducted separately for each sector, with the results averaged to provide a general overview of performance. Additionally, the maximum and minimum values are recorded to illustrate the range of outcomes, highlighting both the best and worst-case scenarios.

Datasets from two countries were available for this analysis, but only the results from Country A are presented. This is for two main reasons: first, the performance of the algorithm was similar in both countries in terms of prediction accuracy and evaluation metrics. Second, the Country B dataset had larger time steps (1 hour), which introduces two issues. It is unlikely that sector usage demand will stay low for a full hour, potentially leading to unnecessary RRU shutdowns. Additionally, turning off the RRU for a full hour inflates energy savings, which could be misleading. While the same issue exists with 15-minute intervals, its impact is less pronounced. Therefore, since the results of the algorithm were similar for both countries, only Country A is shown to ensure the most realistic and representative results are analyzed.

4.1 Impact of Traffic Clustering on Performance

In this analysis, a comparison of the performance of the LSTM model is performed by varying the number of traffic clusters used for model training. The goal is to determine how the number of models, one per cluster, affect the predictive performance of the LSTM across different traffic patterns. By grouping sectors with closely related patterns, we aim to determine whether the benefits of having more clusters (meaning that each model is trained on patterns that are much closer to each other, or, on the extreme, one model for each sector) justify the added computational complexity, as more models need to be trained and maintained.

To conduct this analysis, all algorithms were executed across all available sectors, recording their results. The algorithms were run with a load threshold of 5% and a prediction confidence interval of 0%. Separate models are trained for each cluster, ranging from 1 cluster (a single model for all sectors) to 8 clusters (one model per cluster). Table 4.1 presents the results of this analysis using the same three performance metrics: hit ratio, miss ratio, and accuracy, along with energy saved. The table shows the average results across all sectors, with the maximum and minimum values provided in parentheses for each metric. The same results can be visualized in the bar chart of Figure 4.1.

In some sectors, very few actions were taken because the 5% load threshold was too low, preventing the model from identifying any opportunities for energy-saving actions. This indicates that some sectors experience too much traffic for these efficiencies to be effective when the operator, as in this case, defines a low threshold. This explains why some maximum and minimum values are so bad. Under closer inspection, very few actions are taken in these sectors, sometimes always incorrect, which can end up skewing the average values in some cases.

Table 4.1 - Average Performance of the Algorithm with Different Traffic Clusters.

Model	**Hit Ratio (%)**			**Miss Ratio (%)**			**Accuracy (%)**			**Energy Saved (kWh)**		
	Max.	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**
1 cluster	100	75	0	100	24.9	0	100	65.7	0	14.6	4.16	0.02
2 clusters	100	76.5	0	100	23.5	0	100	61.8	0	14.6	3.98	0.05
4 clusters	100	80.8	45.2	54.8	19.2	0	100	63.9	8	14.6	4.1	0.09
6 clusters	100	78.7	0	100	21.3	0	100	64.5	0	14.7	3.96	0.07
8 clusters	100	76.9	0	100	23.1	0	100	63.3	0	14.6	4.01	0.02

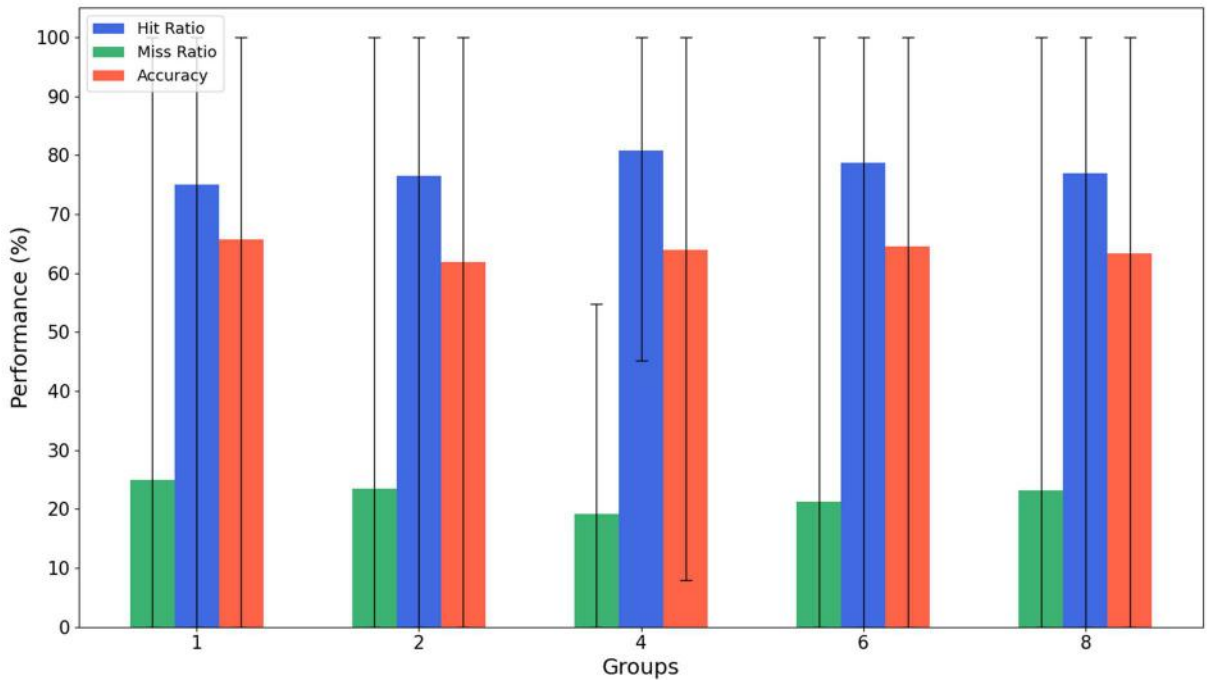


Figure 4.1 - Bar Chart of the Performance of The Algorithm with Varying Traffic Clusters.

We can observe that the 4-group configuration consistently delivers the best results across most metrics, confirming it as the optimal setup for the LSTM model given the available data. The 4-cluster model achieves an average hit ratio of 80.8%, with a perfect maximum of 100% and a strong minimum of 45.2%, demonstrating superior predictive performance, particularly in more challenging scenarios. The miss ratio is also the lowest for this model, averaging 19.2% with a minimum of 0%, indicating fewer missed predictions. Accuracy, while not the highest, remains competitive at an average of 63.9%, with a maximum of 100% and a minimum of 8%. The 1-cluster model reaches a slightly higher accuracy (65.7%), likely because it has all patterns grouped together, which limits its ability to generalize effectively. As a result, the model attempts more actions, leading to a higher accuracy but also a lower hit ratio and a higher amount of energy saved. This suggests that while the model is more active, its predictions are less precise, and the increased number of actions leads to more energy-saving attempts but without the same level of targeted efficiency seen in the 4-cluster model. Energy saved is also strong for the 4-cluster model, with an average of 4.1 kWh, comparable to the other configurations but showing better minimum values. These results align well with the optimal number of clusters identified in Section 3.4.

When considering these three metrics, hit ratio, miss ratio, and accuracy, together, a clear pattern can be observed: the 4-cluster configuration offers best performance across the board. As one deviates from this ideal number of clusters, whether by increasing or decreasing the number of clusters, the performance consistently worsens. This reveals that there is an ideal setup for the number of traffic pattern clusters and that that either oversimplifying with fewer clusters or overcomplicating with more clusters detracts from the effectiveness of the model.

It is important to note that these findings are limited by the small amount of data available for this thesis. With fewer data points, the available data for each cluster decreases as the number of clusters

increases. This can hinder the training of the model or make it more difficult to accurately classify traffic patterns into the right clusters, as the patterns in smaller clusters may closely resemble those in other clusters. These findings could change when more data becomes available, potentially revealing a different ideal number of clusters or even the possibility of achieving even better results, particularly as the scenario where each sector has its own model is approached, when sufficient data is available for training. The limitations imposed by the small dataset may therefore be a factor in why the 4-cluster configuration stands out as the most optimal in this analysis.

4.2 Impact of Confidence Interval of Predictions on Performance

We investigate how varying the confidence interval of predictions impacts the performance of the algorithm. By adjusting the prediction confidence from 0% to 95%, we aim to observe how changes in this parameter affect the behavior of the model across key performance metrics: hit ratio, miss ratio, accuracy and energy saved. The goal is to understand how the sensitivity of the model and caution can be fine-tuned depending on the objectives of the operator. A lower confidence leads to more aggressive decision-making, while a higher confidence makes the model more cautious. This analysis aims to provide insights into how operators can configure the model based on their specific requirements for system performance.

As with the previous analyses, the load threshold remains set at 5%. The results are evaluated across the same three key performance metrics: hit ratio, miss ratio, and accuracy, as shown in Figure 4.2.

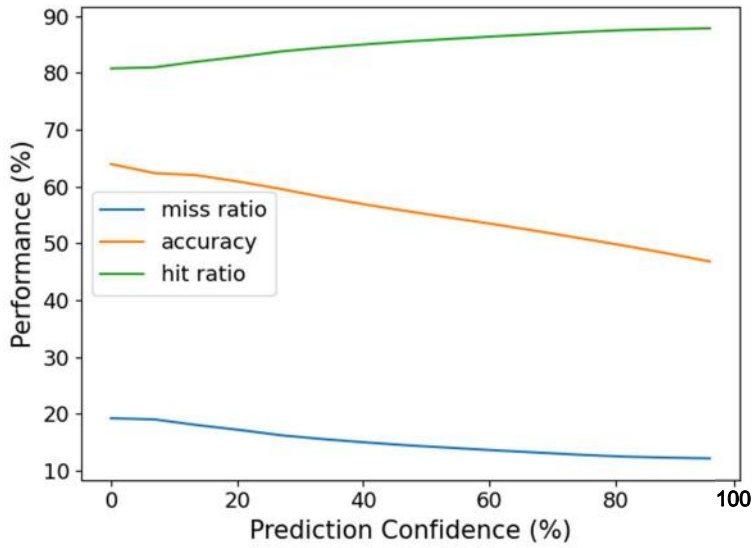


Figure 4.2 - Impact of Prediction Confidence on Hit Ratio, Miss Ratio, and Accuracy.

The results demonstrate a clear trade-off when adjusting the confidence interval of predictions. A higher prediction confidence results in a better hit and miss ratio, which means the model makes more correct predictions, particularly in avoiding false positives. However, this comes at the expense of overall accuracy, as the model becomes more cautious and misses certain opportunities. Conversely, a lower confidence setting yields better overall accuracy but with a worse hit and miss ratio. The energy saved

follows the same trend as accuracy, as while the predictions become more certain, the number of actions taken by the algorithm decreases a lot, decreasing the amount of energy that is saved. As the prediction interval increases, there comes a point where model performance stops improving or only improves marginally, while the model becomes increasingly cautious, causing accuracy and energy savings to drop rapidly. This demonstrates that beyond a certain threshold, increasing the confidence interval is no longer beneficial.

It is concluded that operators can fine-tune the prediction confidence interval to align the performance of the model with their goals: higher confidence for increased certainty in predictions, minimizing the number of incorrect actions taken, or lower confidence for a broader, more aggressive prediction strategy, maximizing the energy saved.

To demonstrate this fine-tuning, confidence intervals are defined with the objective of minimizing incorrect actions, while trying to maintain the energy savings of the algorithm. To find this confidence interval for our data, the same method used to find the optimal number of clusters, the Elbow method is used. For this case, since it is hard to find this point visually, the *KneeLocator()* function of the *kneed* library [Knee24] on version 0.8.5 was used. This function is used to find the point where a significant change in behavior occurs, such as the point at which further increases in one variable result in diminishing returns from another variable. The function first smooths the data, then normalizes it to a unit square. By calculating the differences between the x- and y-values of the normalized curve, it identifies local maxima, which are potential knee points.

We use this method to find the optimal confidence interval for each sector. The results using the confidence intervals obtained are shown in Table 4.2:

Table 4.2 - Performance of the algorithm with Optimized Confidence Interval of Prediction.

Hit Ratio (%)			**Miss Ratio (%)**			**Accuracy (%)**			**Energy Saved (kWh)**		
Max.	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**
100	90.1	50	50	9.9	0	100	46.6	5.1	13.79	3.12	0.065

The results shown in Table 4.2 demonstrate that performance improves across all metrics when using the optimized confidence intervals. Notably, the worst-case scenarios have improved, indicating that the algorithm is now more resilient, even under less favorable conditions. This highlights the value of fine-tuning the confidence intervals, as it leads to more consistent and reliable performance, ensuring that the balance between minimizing incorrect actions and maintaining energy savings is effectively achieved for each sector.

It is important to highlight that the optimal confidence interval is determined by the performance of the algorithm, but the definition of "optimal" depends on the specific goals of the operator. Whether the focus is on a more precise, conservative algorithm or a more aggressive energy-saving approach, the ideal

balance must be set according to the priorities of the operator. The results presented here serve as recommendations, illustrating the trade-offs and potential outcomes for different configurations, but the final decision lies with the objectives of the operator and operational needs.

4.3 Impact of Action Threshold on Performance and Network

In this analysis, the impact of varying the sector load shut-off threshold on the performance of the model is explored. The operator-defined threshold offers additional configurability, allowing actions to be triggered based on predicted PRB load percentages. By adjusting this threshold, network operators can strike a balance between energy savings and maintaining service quality according to their specific needs and priorities.

The prediction confidence is again set to 0%. Figure 4.3 shows how hit ratio, miss ratio, and accuracy change as the threshold increases.

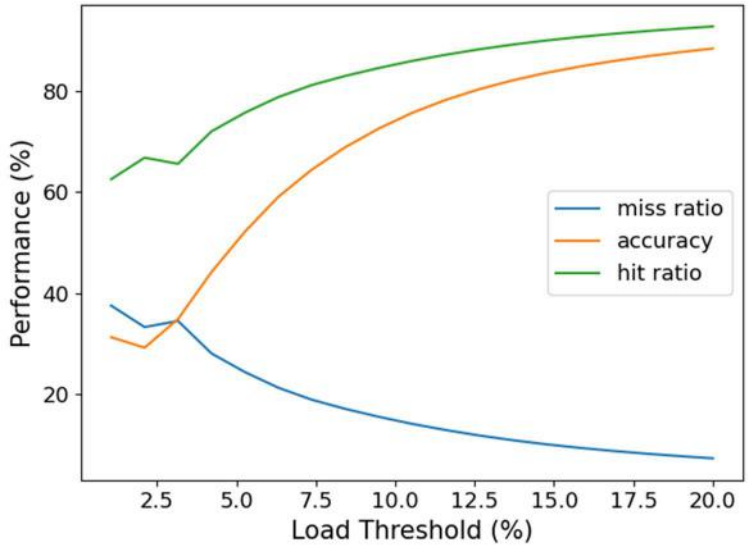


Figure 4.3 - Impact of Load Threshold on Hit Ratio, Miss Ratio, and Accuracy.

We can observe from the graph that as the load threshold increases, the hit ratio steadily rises, while the miss ratio follows the opposite trend and decreases. This behavior can be explained by the fact that, as the threshold becomes closer to the average load of each sector, predictions become easier to make. With higher thresholds, the model is more likely to correctly predict when the PRB load is low enough to trigger an action, resulting in better predictive performance.

The accuracy starts low, and, as more actions are taken with an increasing threshold, the accuracy increases. This rise indicates that higher thresholds make predictions easier and more reliable, with the model correctly predicting actions more frequently, since most of time steps are below the threshold.

While increasing the load threshold increases the amount of correct actions, leading to higher energy savings, these come at a cost. When a sector is turned off, the traffic that would have been taken care of by the turned off sector has to be taken care of by other sites, or technologies present in the current site, which can impact the overall network performance and user experience. The volume of traffic that needs to be managed by alternative sectors or technologies can be quantified from the dataset. As

visualized in Figure 4.4, it can clearly be observed that the trade-off between the energy saved by turning off the RRU and the volume of data that must be handed over to other sectors on average per time step.

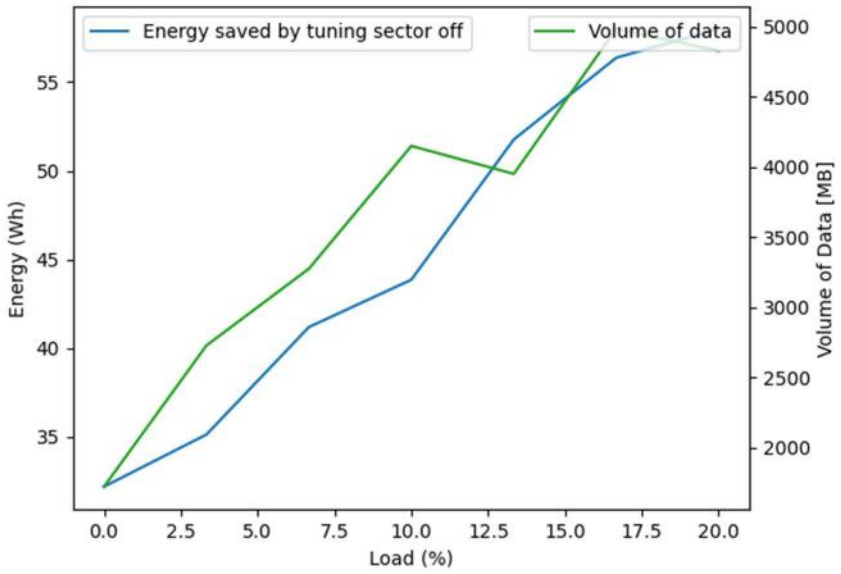


Figure 4.4 - Tradeoff between energy saved and volume of data that needs to be rerouted due to energy saving actions.

As the load threshold increases, both the energy savings and the volume of data increase, which is expected. This trend suggests that higher load thresholds result in more significant energy savings, but also more impact on the network as additional users must be rerouted to other sectors or technologies to handle the increasing traffic demand.

The additional load introduced in other sectors or technologies ultimately leads to increased energy consumption in those sectors. It is assumed that the traffic will be rerouted to 4G networks. To quantify the difference in energy consumption, the antennas of each manufacturer were examined, revealing that without the advanced signal processing present in 5G, 4G antennas consumed roughly 1/4 of the power, with the remaining 3/4 attributed to 5G processing. This estimate represents a worst-case scenario, as the slight increase in load in an already active sector will not cause consumption to rise as significantly as it would if the sector were turned off entirely. Nonetheless, this approximation is used to provide a safe margin for energy calculations.

4.4 Model Robustness Under Unpredictable Traffic Conditions

In this section, an evaluation of the robustness of the LSTM algorithm is done by assessing its performance not only under normal traffic conditions – where the sector traffic follows regular, expected patterns – but also when unexpected traffic occurs. The goal of this analysis is to determine whether the LSTM model can adapt and provide reliable predictions in scenarios where traffic deviates from the patterns it was trained on. By introducing unpredictable traffic variations, one challenges the ability of the model to generalize and handle real-world situations where deviations from the norm are common. This test is crucial to ensure that the algorithm is effective in a dynamic environment, where traffic patterns are not always predictable, and model flexibility becomes critical for performance consistency.

To evaluate how unexpected traffic patterns affect the performance of the model, the focus will be on simulating two types of traffic disruptions: random traffic spikes/drops and event-based traffic changes. For this, the same one day of traffic data for each sector are selected and the model is ran twice: once using regular, unaltered traffic data, and once with simulated disruptions.

1. Random Traffic Spikes and Drops: During specific time windows of the day, five fluctuations are introduced by increasing or decreasing traffic by 100% to simulate random unpredictable surges or dips in the number of users.
2. Event-Based Traffic Changes: We will simulate predictable traffic disruptions by consistently increasing or decreasing traffic over a set block of hours, with the goal of simulating an event or disruption that causes a spike, like a concert or a street manifestation, or a drop in traffic, like very bad weather or a national holiday.

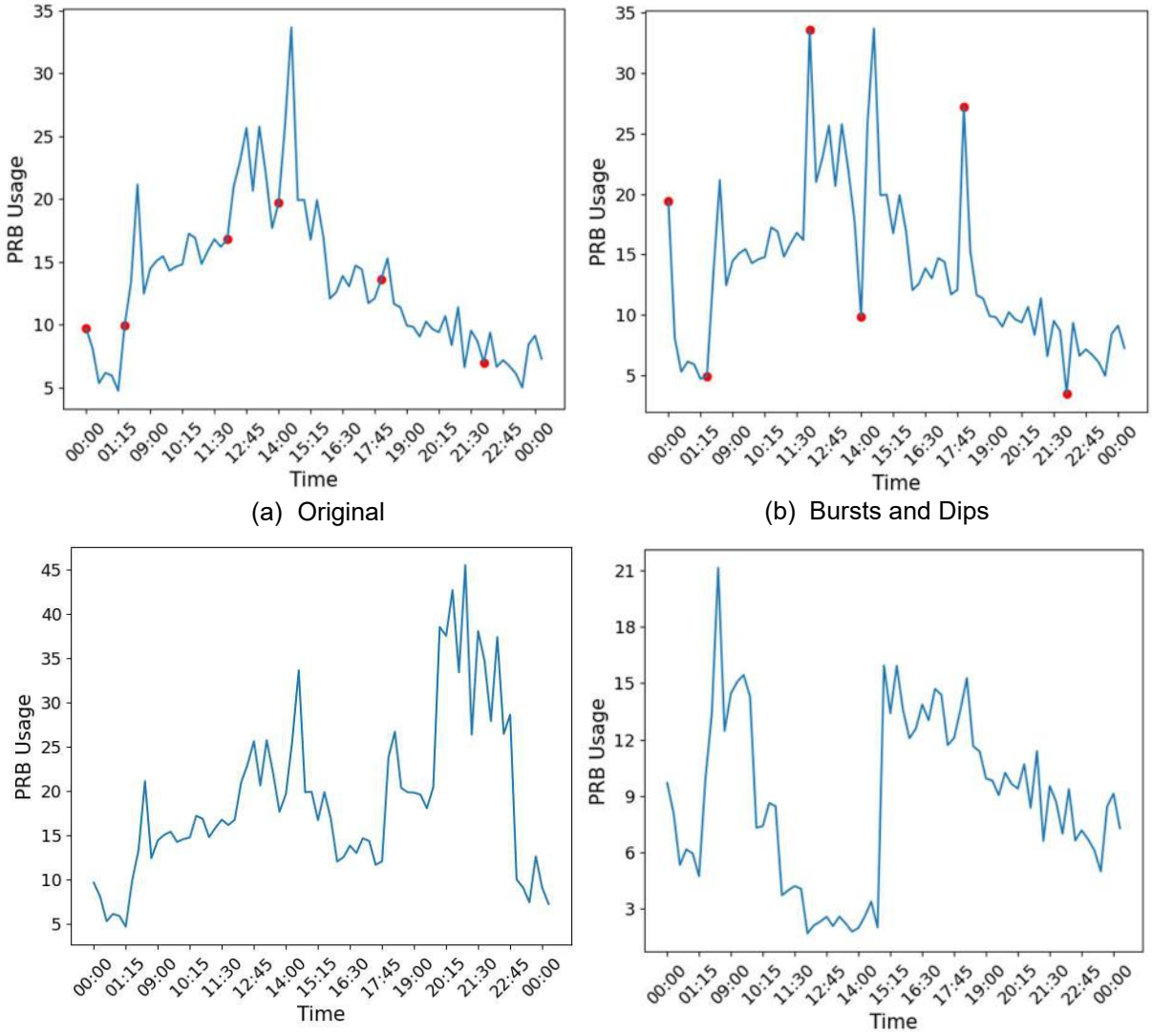
Figure 4.5 represents an example of the four variations of the same day that will be tested to evaluate the LSTM performance of the model under different traffic scenarios.

Figure 4.5 (a) shows the original traffic pattern for a typical day. Figure 4.5 (b) illustrates the bursts and dips in traffic as detailed in point 1., where sudden increases and decreases are introduced at the red points. Figures 4.5 (c) and 4.5 (d) correspond to point 2., with (c) simulating an increase in traffic towards the end of the day, representing an event like a concert, and (d) showing a decreased traffic pattern during peak hours, simulating the effects of a severe weather storm reducing activity.

Figure 4.6 shows the real and predicted traffic usage patterns for the same four different traffic scenarios: (a) the original pattern, (b) bursts and dips in traffic, (c) increased traffic, and (d) decreased traffic.

In Figure 4.6 (b), one observes that the predicted traffic pattern remains close to the original, even when short disruptions (bursts and dips) occur. This is because these disruptions are brief, lasting only for one time step, meaning the LSTM model is unable to predict these sudden changes effectively. Since traffic quickly returns to normal, the predictions of the model revert to what they would have been without the disruption. This indicates that the model does not respond strongly to rapid, temporary fluctuations.

In Figures 4.6 (c) and 4.6 (d), representing increased and decreased traffic, one can see that the predictions of the model are able to follow the disruptions, but with a slight delay on closer inspection. This shows that the model can recognize that a traffic disruption has occurred and attempts to adapt its predictions to follow the trend. However, it does not immediately predict the change and instead lags slightly behind. This delayed adjustment allows for more flexible and responsive actions in handling traffic patterns that deviate from the norm, ensuring the model does not rigidly follow the same pattern every day.

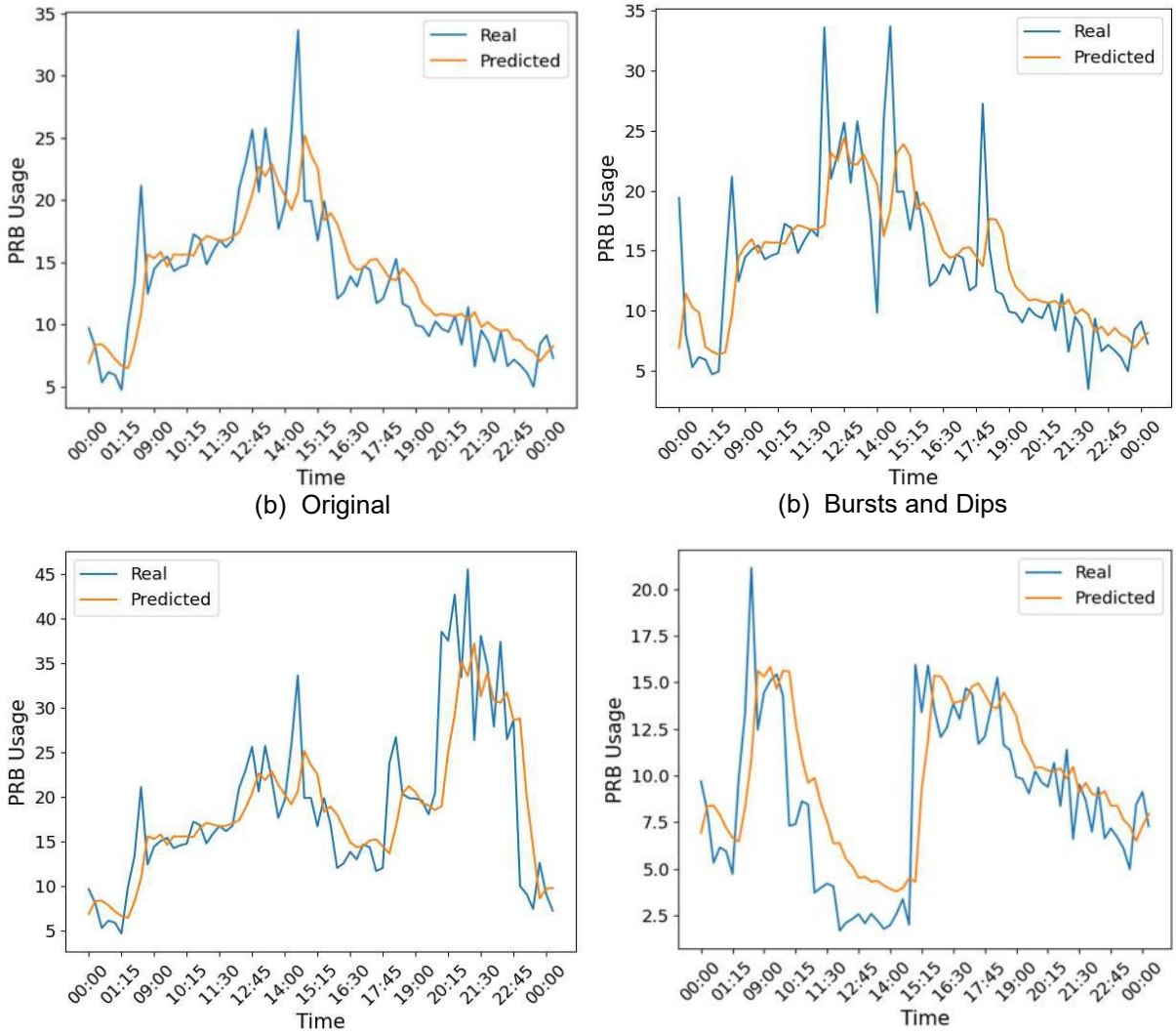


(a) Original (b) Bursts and Dips

(c) Increased Traffic (d) Decrease Traffic

Figure 4.5 - Scenarios Used to Test the Robustness of the Algorithm to Unpredictable Traffic.

As previously discussed, the LSTM model proves to be more effective than traditional statistical methods, primarily because its predictions are closely dependent on previous time steps, allowing it to better capture traffic dynamics. When traffic deviates from normal patterns, the LSTM model adapts without compromising its actions. For instance, in cases where traffic unexpectedly increases, the model ensures that sectors are not permanently shut down, maintaining flexibility and responsiveness. Additionally, even in unpredictable scenarios, the model may still identify opportunities to turn off elements and save energy, demonstrating its robustness and ability to maintain efficiency in dynamic traffic conditions.



(b) Original

(b) Bursts and Dips

(c) Increased Traffic

(d) Decrease Traffic

Figure 4.6 – Predictions of the Model on Different Traffic Scenarios.

4.5 Evaluating Algorithm Generalization: Performance in Trained vs. Untrained Sectors

In this section, the performance of the algorithm is assessed by comparing its predictive capabilities across sectors that differ in their training status. The objective is to investigate how the algorithm performs when applied to a sector that has been included in the training dataset versus one that is entirely new and unseen. By evaluating the ability of the model to generalize to an unfamiliar site, one can verify the robustness of the model and adaptability in real-world applications, where new sites would not have their historic data included in the training data. Table 4.3 shows the comparison between the performance of the optimized model with an untrained sector.

Table 4.3 - Performance Comparison of the Optimized LSTM with an Untrained Sector.

Model	Hit Ratio (%)	Miss Ratio (%)	Accuracy (%)	Energy Saved (kWh)
Untrained sector	91.1	8.9	42.1	7.08
LSTM	90.1	9.9	46.6	3.12

The untrained sector demonstrates superior hit and miss ratios compared to the average results of the LSTM model, performing among the best outputs of the LSTM. However, it also exhibits significantly lower accuracy, positioning it on the lower end of the accuracy spectrum when compared to other LSTM results. These statistics indicate that the untrained sector predictions are more cautious, with the model predominantly taking correct actions but consequently missing numerous available opportunities to act.

This cautious behavior can be attributed to the fact that the untrained sector historical data was not included in the training process. As a result, the predictions of the model for this sector are more generalized, lacking the nuanced understanding of specific traffic patterns that it could leverage if it had been exposed to the relevant training data. Interestingly, since the patterns in the untrained sector are similar to those used during training, the model is still able to make reasonably good predictions, resulting in performance metrics that align well with those of other sectors. This suggests that the LSTM can generalize effectively to some degree when faced with familiar patterns, even in the absence of direct training data.

4.6 Performance Comparison with Baseline Models

In analysis, the performance of the LSTM model is compared against three baseline models – naive, seasonal naive, and moving average. The primary goal is to evaluate whether the LSTM demonstrates superior predictive abilities. Since the baseline models rely solely on previous values and lack predictive capabilities, the LSTM must outperform them to prove its effectiveness. Failing to do so would indicate that our model is inferior to these simpler approaches, which would suggest that its predictive abilities are inadequate.

To conduct this analysis, all algorithms were executed using the LSTM and baseline models as the prediction models across all available sectors, recording their results. The algorithms were run with a load threshold of 5% and a prediction confidence of 0%. The previously obtained results for the LSTM with optimized confidence intervals is also included. Table 4.4 shows the results of the tests using three performance metrics, hit ratio, miss ratio, and accuracy, along with the energy saved. The table shows the mean results across all sectors, with the maximum and minimum values indicated in parentheses.

Table 4.4 - Performance of the Algorithm Comparison with Baseline Models.

Model	Hit Ratio (%)			Miss Ratio (%)			Accuracy (%)			Energy Saved (kWh)		
	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.
Naive	100	72.1	20	80	27.9	0	100	67.3	10	14.46	4.27	0.096
Seasonal Naive	100	72.8	28.5	71.4	27.2	0	100	67.2	15.4	14.46	4.27	0.067
Moving Average	100	67.1	0	100	32.9	1.79	100	63.5	0	14.66	4.3	0.1
LSTM	100	80.8	45.2	54.8	19.2	0	100	63.9	8	14.61	4.1	0.092
Optimized LSTM	100	90.1	50	50	9.9	0	100	46.6	5.1	13.79	3.12	0.065

The LSTM model achieves the highest average hit ratio at 80.8%, with a perfect maximum score of 100% and a minimum score of 45.2%, outperforming all baseline models. This indicates that the LSTM is significantly better at making correct predictions that lead to the desired action while demonstrating greater consistency in less favorable scenarios.

Correspondingly, the LSTM has a lower average miss ratio at 19.2%, along with lower maximum and minimum percentages compared to the baseline models. This reinforces the reliability of the LSTM in minimizing incorrect actions, with a minimum miss ratio of 0%, highlighting its potential for error-free performance in certain cases without requiring further tuning.

In terms of accuracy, the LSTM shows an average of 63.9%. However, this value is closely aligned with those of the baseline models, which exhibit slightly higher accuracy rates due to their more aggressive approach in capturing extreme variations in the target. While these aggressive models identify more sleep opportunities, they also incur a significantly higher number of incorrect actions. This result indicates that, although the LSTM excels in making correct predictions, it does not capitalize on all available opportunities for action, demonstrating the tradeoff between more aggressive and more conservative models.

Contrary to the past findings, the LSTM has the lowest average energy saved. The reason for this comes from how energy savings are calculated. Every time the sector is turned off, the energy saved with these actions is calculated and summed, regardless of it being a correct action or not. Consequently, because the baseline models have a significantly higher average miss rate, the energy saved from these incorrect actions accumulates to a greater total. The problem with these incorrect actions is that not only the intended operation of the sector is disturbed but also require rerouting more traffic than anticipated to other sites or technologies. Given this context, it is reasonable to conclude that the energy saved by the

LSTM model, although smaller, is a more accurate reflection of the results of an “ideal” model, as it aligns better with the intended network operation.

When comparing these results with the optimized LSTM, it becomes evident that while the model exhibits greater correctness in its actions, it also adopts a more cautious approach, as reflected in the lower accuracy and energy savings. Despite this, such behavior aligns with the intended design of the model, where minimizing incorrect actions is crucial for maintaining the network functionality goals. This cautiousness ensures that the model prioritizes reliability over aggressive performance, ultimately supporting the overall efficiency and stability of the network.

4.7 Economic and Environmental Impact Assessment of the Final Model

Using the final model with the previously obtained “optimal” confidence intervals with a 5% threshold, the energy saved is taken to analyze the economic and environmental impact of the proposed algorithm. To obtain these values, publicly available data is used to evaluate the cost savings associated with reduced energy consumption and the reduction of CO2 emissions resulting from the energy savings achieved by the model. The following data is used for the calculation of the results:

- On average, the algorithm saved 3.12 kWh per sector in one week.
- The average kWh price in Country A in 2024 is 0.1644€.
- The average CO2 emissions in Country A in 2023 were 165.5 g CO2 per kWh.
- Vodafone has 5093 available sites in Country A as of the second quarter of 2023.

The following results are obtained for each sector on average:

- Energy Saved: 162.7 kWh per year per sector.
- Money Saved: 26.74 € per year per sector.
- CO2 Emissions Saved: 26.9 kg per year per sector.

To obtain an overview of the potential best-case results, it is assumed that the average outcomes of the algorithm are representative of the entire population of BSs. The number of available sites, including those that have not yet fully transitioned to 5G, is used to estimate the economic and environmental impact. This approach allows us to project what the overall impact could be once all sites eventually adopt 5G technology. The results from the use of the algorithm are presented in Table 4.5:

Table 4.5 - Economic and Environmental Impact of the Algorithm.

Duration	Total Energy Saved (kWh)	Total Money Saved (€)	Total CO2 Emissions Saved (tons)
Week	47 679	7 838	7.9
Month	204 302	33 587	33.8
Year	2 451 624	403 047	405.7

Chapter 5

Conclusions

The primary goal of this thesis was to develop an algorithm that optimizes energy consumption in 5G networks by using traffic predictions to implement energy-saving actions without compromising network performance. This goal was successfully achieved, demonstrating the potential for significant energy savings, particularly during low-traffic periods. The results indicate that Machine Learning powered traffic predictions can effectively enable proactive energy-saving measures, contributing to more efficient network operations.

Chapter 1 provides a broad overview of the thesis topic, outlining the key issues surrounding energy efficiency in mobile networks. It highlights the motivation behind the research, the main objectives, and the relevance of the work. The focus is on developing and implementing machine learning techniques to improve energy efficiency in 4G and 5G networks, particularly targeting enhancements to the Radio Access Network.

Chapter 2 provides a comprehensive review of the fundamental concepts relevant to this thesis. It begins with an overview of the architecture and radio interfaces of 4G and 5G networks, emphasizing the importance of understanding these aspects for identifying potential energy efficiencies. This is followed by a detailed analysis of energy consumption in both network types, pinpointing areas where improvements can be made and outlining the metrics used to evaluate energy efficiency in mobile networks. The chapter also explores the services introduced by 5G, along with their performance requirements, to establish a baseline for network efficiency without compromising service quality. Additionally, an introduction to machine learning is provided, covering various learning processes and algorithms, with a focus on those that will be used in the thesis development. The chapter concludes with a review of the state of the art, summarizing previous work in the field and their findings.

Chapter 3 builds on the foundational concepts introduced in Chapter 2 and applies them to both understand the available data and develop the proposed algorithm. The chapter begins by outlining the different components that make up the final algorithm, setting the stage for a detailed exploration of each. First, the datasets provided are discussed in depth, focusing on their structure and content. Following this, an exploratory data analysis is conducted with the aim of fully understanding the characteristics of the available data. This process involves examining feature correlations, selecting relevant features, and using various data visualization techniques to better comprehend underlying patterns. Once the data is thoroughly understood, the chapter moves on to explain how each sector of the network will be clustered. This clustering is essential to reduce the complexity of the model while maintaining performance, allowing the algorithm to generalize better across different network sectors. The chapter then details the LSTM model that will be used for traffic prediction. The data preprocessing steps necessary for this model are also described. Additionally, a confidence interval is introduced for the predictions of the model, which enhances the accuracy and reliability of the predictions by providing a measure of uncertainty. Finally, the complete algorithm is presented, tying together all the prior steps. The actions of the algorithm are explained, including the benefits it aims to achieve, the potential drawbacks it must address, and the test metrics that will be used to evaluate its performance. This chapter serves as the foundation for the practical implementation and testing of the energy-saving measures developed in the thesis.

Chapter 4 evaluates the algorithm developed in the previous chapter, examining the influence of the various customization variables on its performance. It conducts tests under different scenarios involving unpredictable traffic, simulating disruptions to normal operations. Additionally, the chapter provides a comparative analysis against baseline models to assess the effectiveness of the algorithm. Finally, it incorporates an assessment of the economic and environmental impacts, underscoring the broader implications of the proposed solution.

Several positive outcomes were identified through this work. First, there is a clear opportunity for time-based efficiencies in 5G systems, especially during periods of low or no traffic. The analysis showed that by predicting traffic accurately, it is possible to act preemptively, turning off or reducing the power of network components during these times, leading to tangible energy savings. This approach leverages ML models to improve energy efficiency without impacting the quality of service. Additionally, the tests performed with introduced traffic fluctuations demonstrate that the model effectively understands these changes, tracking them closely without significant performance degradation. This adaptability is a key benefit of using machine learning, specifically the LSTM model, which captures the seasonality of the data while giving meaningful weight to past values. As a result, the model remains responsive to varying conditions, enhancing its overall robustness.

Despite the promising results, our model faced several limitations, primarily due to the granularity of the available data. Since the datasets consist of time steps that are either 15 minutes or 1 hour long, it is impossible to identify precise moments with no traffic. This limitation affects both the actions that can be taken and how their impact on the network can be accurately measured. The initial idea was to sequentially turn off specific components of the base station during periods with no active users, thus improving energy efficiency without affecting user experience. However, with large time steps, it is challenging to detect those exact windows of no traffic, limiting the effectiveness of the energy-saving actions one can implement. Additionally, with larger time steps, short-term fluctuations and subtle patterns in traffic are smoothed out or lost, making it difficult for the model to capture smaller variations. This limits the ability of the model to accurately anticipate rapid changes in traffic, reducing its overall precision and limiting the ability to precisely predict when network components could be powered down without disrupting service.

Moreover, with more granular data, one could also leverage Reinforcement Learning (RL) to optimize the energy-saving algorithm within a more realistic network simulation. RL would allow the model to dynamically adjust actions based on predicted no-traffic periods while considering network latency and QoS constraints. This would enable a more sophisticated optimization process, where the system intelligently balances the depth of power-saving actions with the need to maintain network performance, achieving a better overall efficiency.

This thesis has made several contributions toward more efficient 5G network management. It highlights that additional energy-saving techniques can be implemented today, using traffic predictions to turn off or reduce network components based on expected traffic patterns. While this thesis focused on time-based efficiencies, the approach can also be extended to other dimensions such as spatial and

frequency, where certain unused frequencies or certain antennas from MIMO covering unoccupied areas can be temporarily turned off, further enhancing energy efficiency in 5G networks.

Future work should focus on obtaining more detailed datasets with finer time granularity and richer information on network performance and QoS. This would allow for improved traffic predictions by incorporating more complex machine learning models, which can better capture short-term traffic fluctuations and patterns. Additionally, with the availability of more comprehensive data, a realistic simulation of network traffic, performance, and QoS could be implemented. This would enable the use of reinforcement learning (RL) to optimize the system dynamically, allowing for greater energy savings while minimizing disruptions and added latency based on real-time network needs.

Moreover, it is worth mentioning that user-side data could also be considered in future work. Data that the network does not currently access, such as battery usage, user location, terminal model, and capabilities, can provide valuable insights. Integrating this information may enhance the ML model's performance by enabling it to better understand user behavior and demand patterns, ultimately leading to more accurate traffic predictions and energy-saving strategies.

Furthermore, since information on sector usage is lost when the RRU is off, predictions must fill in these time steps. This could be achieved using the developed LSTM or another predictive model, although if multiple time steps are missed, predictions may degrade. To address this, leveraging external data—particularly from other technologies and frequencies on-site—could allow for a more accurate general demand assessment at that location.

In conclusion, this thesis has demonstrated the potential for significant energy savings in 5G networks through the use of traffic predictions and targeted energy-saving actions. There remains considerable room for improvement, particularly in terms of action optimization, predictive accuracy and improved training data, but the findings provide a strong foundation for future research and implementation.

Annex A

Dataset Characteristics

This annex provides a comprehensive overview of the dataset and its features, detailing the specific locations of the sites included in the dataset, along with their radio characteristics and antenna manufacturers. It aims to present an in-depth understanding of the data attributes, facilitating better insights into the underlying infrastructure and its capabilities.

This annex is confidential.

References

[3GPP02] 3GPP, *Universal Mobile Telecommunications System (UMTS); Quality of Service (QoS) concept and architecture (Release 5)*, Report TS 23.107, Sophia Antipolis, France, Mar. 2002, Available: https://www.etsi.org/deliver/etsi_ts/123100_123199/123107/05.04.00_60/ts_123107v050400p.pdf.

[3GPP10] 3GPP, *Telecommunication management; Study on Energy Savings Management (ESM) (Release 10)*, Report TR 32.826, Sophia Antipolis, France, Mar. 2010, Available: https://www.3gpp.org/ftp/Specs/archive/32_series/32.826/32826-a00.zip.

[3GPP16] 3GPP, *Technical Specification Group Services and System Aspects; Feasibility Study on New Services and Markets Technology Enablers; Stage 1 (Release 14)*, Report TR 22.891, Ver. 14.2.0, Sophia Antipolis, France, Set. 2016, Available: https://www.3gpp.org/ftp/Specs/archive/22_series/22.891/22891-e20.zip.

[3GPP19] 3GPP, *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; (Release 15)*, Report TR 21.915, Ver. 15.0.0, Sophia Antipolis, France, Oct. 2019, Available: https://www.etsi.org/deliver/etsi_tr/121900_121999/121915/15.00.00_60/tr_121915v150000p.pdf.

[3GPP20] 3GPP, *5G; LTE; Management and orchestration; Energy efficiency of 5G (Release 16),* Report TS 28.310, Ver. 16.1.0, Sophia Antipolis, France, Aug. 2020, Available: https://www.etsi.org/deliver/etsi_ts/128300_128399/128310/16.01.00_60/ts_128310v160100p.pdf.

[3GPP23] 3GPP, *5G; System architecture for the 5G System (5GS) (Release 17)*, Report TS 23.501, Ver. 17.8.0, Sophia Antipolis, France, Apr. 2023, Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/17.08.00_60/ts_123501v170800p.pdf.

[AGDG12] Gunther Auer, Vito Giannini, Claude Desset, István Gódor, Per Skillermark, Magnus Olsson, Muhammad Ali Imran, Dario Sabella, Manuel J. Gonzalez, Oliver Blume, A Lbrecht Fehske, “How much energy is needed to run a wireless network?”, *IEEE Wireless Communications*, Vol: 18, No. 5, pp. 40-49, July 2012, Available: https://www.researchgate.net/publication/234076347_How_much_energy_is_needed_to_run_a_wireless_network.

[AKSG08] Alexander Kraskov, Harald Stoegbauer, Peter Grassberger, “*Estimating Mutual Information*”, John-von-Neumann Institute for Computing, Julich, Germany, Feb. 2008, Available: https://arxiv.org/pdf/cond-mat/0305641.

[BGSE15] Gérard Biau, Erwan Scornet, “*A Random Forest Guided Tour*”, UPMC Univ Paris, Paris, France, Nov. 2015, Available: https://arxiv.org/pdf/1511.05741.

[BKPS93] Ben Krose, Patrick van der Smagt, “An introduction to neural networks”, *Journal of*

Computer Science, Vol 48, Jan. 1993, Available: https://www.researchgate.net/publication/272832321_An_introduction_to_neural_networks.

[BPDR10] Peter Brockwell, Richard A. Davis, *Introduction to Time Series and Forecasting*, Springer Texts in Statistics, Second edition, New York, USA, 2010.

[Brei01] Leo Breiman, “Random Forests”, *Machine Learning*, Vol. 45, Oct. 2001, pp. 5-32, Available: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf.

[BVJA19] Venkatesh, Anuradha, “A Review of Feature Selection and Its Methods”, *Cybernetics and Information Technologies*, Vol. 19, No. 1, Feb. 2019, pp. 3-26, Available: https://doi.org/10.2478/cait-2019-0001.

[CCMT90] Robert Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning, “STL: A Seasonal-Trend Decomposition Procedure Based on LOESS”, *Journal of Official Statistics*, Vol. 6, No. 1, 1990, pp. 3-73, Available: https://www.wessa.net/download/stl.pdf.

[CGSS20] Xinxin Chai, Hui Gao, Ji Sun, Xin Su, Tiejun Lv, Jie Zeng, “Reinforcement Learning Based Antenna Selection in User-Centric Massive MIMO”, *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, May 2020, Available: https://doi.org/10.1109/VTC2020-Spring48590.2020.9129108.

[CHAC20] Caroline Gabriel, Hansang (Andy), Andrew Chern, *Green 5g: building a sustainable world*, Huawei, Analysis Mason, London, UK, Aug. 2020, Available: https://www.huawei.com/en/public-policy/green-5g-building-a-sustainable-world.

[Chok10] Nian Shong Chok, “*Pearson’s versus Spearman’s and Kendall’s correlation coefficients for continuous data*”, Winona State University, Minnesota, USA, 2010, Available: http://d-scholarship.pitt.edu/8056/1/Chokns_etd2010.pdf.

[CKYY10] Tao Chen, Haesik Kim, Yang Yang, "Energy efficiency metrics for green wireless communications", *2010 International Conference on Wireless Communications & Signal Processing (WCSP)*, Suzhou, China, pp. 1-6, Nov. 2010, Available: https://ieeexplore.ieee.org/document/5633634.

[CTHT24] Tianqi Chen, Tong He, “*xgboost: eXtreme Gradient Boosting*”, July 2024, Available: https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf.

[DKAK12] Dongare, Kharde, Amit Kachare, “Introduction to Artificial Neural Network”, *International Journal of Engineering and Innovative Technology*, Vol.2, No.1, July 2012, pp. 189-194, Available: https://api.semanticscholar.org/CorpusID:212457035.

[DSNK17] Diksha Sharma, Neeraj Kumar, “A Review on Machine Learning Algorithms, Tasks and Applications”, *International Journal of Advanced Research in Computer Engineering & Technology*, Vol. 6, No. 10, Oct. 2017, pp. 1548-1552, Available: https://www.researchgate.net/publication/320609700_A_Review_on_Machine_Learning_Algorithms_Tasks_and_Applications.

[Eric23] Ericsson, *Ericsson Mobility Report*, Nov. 2023, Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[ETPZ09] Pablo Estevez; Michel Tesmer; Claudio A. Perez; Jacek M. Zurada, "Normalized Mutual Information Feature Selection", *IEEE Transactions on Neural Networks*, Vol.20, No.2, Feb. 2009, pp. 189-201, Available: https://doi.org/10.1109/TNN.2008.2005601.

[HBVB11] Ziaul Hasan, Hamidreza Boostanimehr, Vijay K. Bhargava, "Green Cellular Networks: A Survey, Some Research Issues and Challenges", *IEEE communications surveys & tutorials*, Vol. 13, No. 4, Nov. 2011, pp. 524-540, Available: https://ieeexplore.ieee.org/document/6065681.

[HHGS16] Hado van Hasselt, Arthur Guez, David Silver, "Deep Reinforcement Learning with Double Q-Learning", *AAAI Conference on Artificial Intelligence*, Ver. 30, No.1, Arizona, USA, 2016, pp. 2094- 2100, Available: https://doi.org/10.1609/aaai.v30i1.10295.

[HLGC24] Rachida Hachache, Mourad Labrahmi, António Grilo, Abdelaali Chaoub, Rachid Bennani, Ahmed Tamtaoui, Brahim Lakssir, "Energy Load Forecasting Techniques in Smart Grids: A Cross-Country Comparative Analysis", *Energies*, Vol 17, No. 10, May 2024, Available: https://doi.org/10.3390/en17102251.

[ICSJ19] Mohaiminul Islam, Guorong Chen, Shangzhu Jin, "An Overview of Neural Network", *American Journal of Neural Networks and Applications*, Vol. 5, No. 1, June 2019, pp. 7-11, Available: https://doi.org/10.11648/j.ajnna.20190501.12.

[ILSL23] Toufiqul Islam, Daewon Lee, Seau Sian Lim, "Enabling Network Power Savings in 5G-Advanced and Beyond", *IEEE journal on selected areas in communications*, Vol. 41, No. 6, June 2023, Available: https://ieeexplore.ieee.org/document/10121451.

[ITUR15] ITU-R, *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*, Recommendation ITU-R M.2083-0, Set. 2015.

[JKHK19] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Jong Wook Kim, "Q-Learning Algorithms: A Comprehensive Classification and Applications", *IEEE Access*, Vol. 7, 2019, pp. 133653- 133667, Available: https://ieeexplore.ieee.org/document/8836506.

[Kera24] *Keras version 3.5.0 documentation*, Available: https://keras.io/api/.

[KHSM21] Emanuel Kolta, Tim Hatt, Steven Moore, *Going green: benchmarking the energy efficiency of mobile*, GSMA Intelligence, London, UK, June 2021, Available: https://data.gsmaintelligence.com/api-web/v2/research-file-download?id=60621137&file=300621-Going-Green-efficiency-mobile.pdf.

[Knee24] *Kneed version 0.8.5 documentation*, Available: https://kneed.readthedocs.io/en/stable/.

[KTMP13] Trupti Kodinariya, Prashant Makwana, "Review on determining number of Cluster in K-Means Clustering", *International Journal of Advance Research in Computer Science and*

Management Studies, Vol. 1, No. 6, Nov. 2013, Available: https://www.researchgate.net/profile/Trupti-Kodinariya/publication/313554124_Review_on_Determining_of_Cluster_in_K-means_Clustering/links/5789fda408ae59aa667931d2/Review-on-Determining-of-Cluster-in-K-means-Clustering.pdf.

[Laun21] Frédéric Launay, *NG-RAN and 5G-NR; 5G Radio Access Network and Radio Interface*, ISTE Ltd, Great Britain, 2021.

[LCWM10] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, Huan Liu, "Feature Selection: A Data Perspective", *ACM Comput. Surv*., Vol.9, No.4, March 2010, Available: http://dx.doi.org/10.1145/0000000.0000000.

[LLLC17] Leo Liberti, Carlile Lavor, "*Euclidean Distance Geometry*", Springer Undergraduate Texts in Mathematics and Technology, Springer, Switzerland, 2017, Available: https://link.springer.com/content/pdf/10.1007/978-3-319-60792-4.pdf.

[Mahe20] Batta Mahesh, "Machine Learning Algorithms - A Review", *International Journal of Science and Research*, Vol. 9, No.1, Jan. 2020, pp. 381-386, Available: https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-A_Review.

[Marq22] Jorge Marques, *Machine Learning*, Lecture Notes, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, Oct. 2022.

[MCHM23] Mohamed-Amine Chadia, Hajar Mousannifa, *Understanding Reinforcement Learning Algorithms: The Progress from Basic Q-learning to Proximal Policy Optimization*, University of Cadi Ayyad, Marrakech, Morocco, Mar. 2023, Available: https://arxiv.org/abs/2304.00026.

[MCIT15] Ministry of Communications and Information Technology, *Energy Efficiency testing of various Telecom equipment and Green Telecom*, Available: https://tec.gov.in/pdf/Studypaper/final%20LATEST%20ENERGY%20EFFICIENCY22MARCH.pdf.

[MHPK21] Marcin Hoffmann, Pawel Kryszkiewicz, "Reinforcement Learning for Energy-Efficient 5G Massive MIMO: Intelligent Antenna Switching", *IEEE Access*, Vol. 9, Sept. 2021, pp. 130329-130339, Available: https://doi.org/10.1109/ACCESS.2021.3113461.

[NGMN15] NGMN, *NGMN White Paper*, Version 1, Feb. 2015, Available: https://ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf.

[PBRR11] Aruna Prem Bianzino, Anand Kishore Raju, Dario Rossi, "Apples-to-Apples: A Framework Analysis for Energy-Efficiency in Networks", *ACM SIGMETRICS Performance Evaluation Review*, Vol. 32, No. 3, Jan. 2011, pp. 81-85, Available: https://dl.acm.org/doi/10.1145/1925019.1925036.

[PDPX22] David López-Pérez, Antonio De Domenico, Nicola Piovesan, Geng Xinli, Harvey Bao, Song Qitao, Mérouane Debbah, "A Survey on 5G Radio Access Network Energy Efficiency: Massive MIMO, Lean Carrier Design, Sleep Modes, and Machine Learning", *IEEE communications surveys & tutorials*, Vol. 24, N0. 1, First quarter 2022, pp. 653-697,

Available: https://ieeexplore.ieee.org/document/9678321.

[PIRP22] Anita Patil, Sridhar Iyer, Rahul Avantha Pandya, *Machine Learning Algorithms for 6G Wireless Networks: A Survey*, June 2022. Available: https://arxiv.org/abs/2203.08429.

[PLST16] Ngoc Phuc Le, Farzad Safaei, Le Chung Tran, "Antenna Selection Strategies for MIMO-OFDM Wireless Systems: An Energy Efficiency Perspective", *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 4, April 2016, pp. 2048-2062, Available: https://doi.org/10.1109/TVT.2015.2428815.

[PPDG22] Nicola Piovesan, David López-Pérez, Antonio De Domenico, Xinli Geng, Harvey Bao, and Mérouane Debbah, "Machine Learning and Analytical Power Consumption Models for 5G Base Stations", *IEEE Communications Magazine*, Vol. 16, No. 10, Oct. 2022, Available: https://ieeexplore.ieee.org/document/9928089.

[SBKG16] Shuvabrata Bandopadhaya, Kishore Kumar Gupta, Adaptive antenna selection technique for MIMO system, *2016 3rd International Conference on Signal Processing and Integrated Networks*, Noida, India, Feb. 2016, Available: https://doi.org/10.1109/SPIN.2016.7566735.

[SCAG19] Fatma Ezzahra Salem, Tijani Chahed, Eitan Altman, Azeddine Gati, Zwi Altman, "Optimal Policies of Advanced Sleep Modes for Energy-Efficient 5G networks", *2019 IEEE 18th International Symposium on Network Computing and Applications*, Cambridge, MA, USA, Sep. 2019, Available: https://doi.org/10.1109/NCA.2019.8935062.

[Scik24] *Scikit-learn version 1.5.1 documentation*, Available: https://scikit-learn.org/1.5/.

[SGAC17] Fatma Ezzahra Salem, Azeddine Gati, Zwi Altman, Tijani Chahed, "Advanced Sleep Modes and their impact on flow-level performance of 5G networks", *IEEE Conference on Vehicular Technology*, Châtillon, France, Sep. 2017, Available: https://ieeexplore.ieee.org/document/8288125.

[Sher20] Alex Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network", *Physica D: Nonlinear Phenomena*, Vol. 404, March 2020, Available: https://arxiv.org/pdf/1808.03314.

[SMRH22] Ibrahim Salah, M. Mourad Mabrook, Kamel Hussein Rahouma, Aziza I. Hussein, "Energy efficiency optimization in adaptive massive MIMO networks for 5G applications using genetic algorithm", *Opt Quant Electron*, Vol. 54, Jan. 2022, Available: https://doi.org/10.1007/s11082-021-03507-5.

[SRBM21] Shurdi, Ruci, Biberaj & Mesi (2021), "5G Energy Efficiency Overview", *European Scientific Journal,* Vol. 17, No. 3, Jan. 2021, pp. 315-327, Available: https://doi.org/10.19044/esj.2021.v17n3p315.

[SSSS20] Siddharth Sharma, Simone Sharma, "Activation Functions In Neural Networks", *International Journal of Engineering Applied Sciences and Technology*, Vol. 4, No. 12, April 2020, pp. 310-316, Available: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[Stats24] *Statsmodels version 0.14.2 documentation*, Available: https://www.statsmodels.org/v0.14.4/index.html.

[TaPe19] Petroc Taylor, *Number of smartphone mobile network subscriptions worldwide from 2016 to 2022, with forecasts from 2023 to 2028*, Jul. 2019, Statista, Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[Tens24] *TensorFlow version 2.17.0 documentation*, Available: https://www.tensorflow.org/api_docs/python/tf.

[THEK20] Tim Hatt, Emanuel Kolta, *5G energy efficiencies Green is the new black*, GSMA Intelligence, Nov. 2020, Available: https://data.gsmaintelligence.com/signin?returnPath=/research/research/research-2020/5g-energy-efficiencies-green-is-the-new-black.

[TWTH01] Robert Tibshirani, Guenther Walther, Trevor Hastie, "Estimating the number of clusters in a data set via the gap statistic.", *Journal of the Royal Statistical Society*, pp. 411-423, 2001.

[TWYZ23] Tong Wu, Yulong Zou, "Energy Efficiency Optimization in Adaptive Transmit Antenna Selection Systems with Limited Feedback", *IEEE Internet of Things Journal*, Vol. 10, No. 2, Sept. 2022, pp. 1248-1258, Available: https://doi.org/10.1109/JIOT.2022.3206460.

[VaLS23] Lionel Sujay Vailshery, *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030*, Jul. 2023, Statista, Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[YLHJ17] Heejung Yu, Howon Lee, Hongbeom Jeon, "What is 5G? Emerging 5G Mobile Services and Network Requirements", *Sustainability*, Vol. 9, No. 10, Oct.2017, Available: https://doi.org/10.3390/su9101848.