UNIVERSITY OF THESSALY
Electrical & Computer Engineering Dept.

Diploma Thesis

Design & Implementation of Service Placement Algorithms in Multi-Access Edge Computing Setups for V2X Communications

Διπλωματική Εργασία

Σχεδιασμός και Υλοποίηση Αλγορίθμων Χωροθέτησης Υπηρεσιών στην Άκρη του Δικτύου για Επικοινωνίες Κινητών Οχημάτων

Author: Christos Nanis

(cnanis@uth.gr)

Supervisor

Athanasios Korakis

Committee Members

Fotios Plessas

Eleftherios Tsoukalas

A thesis submitted in partial fulfillment of the requirements for the degree of Diploma Thesis/Integrated Master

Volos, Greece, September 2020

Abstract

Paving our way towards next generation mobile networks (5G), we first need to consider which gradually - emerging problems, currently used technologies suffer from, to be addressed properly through efficient Network Architecture Design and Implementation. In order to offer a ubiquitous and personalized experience to the mobile Internet user, it is of extreme importance to max out on currently used technologies. This means enhancing and building on top, testing and fine-tuning, so as their boundaries can be reached and then surpassed, along with the standardization of newly introduced concepts (e.g. mmWave, beamforming, extensive disaggregation and virtualization of RAN components, etc.).

A wide range of envisioned or at an early development stage applications, have each formed their standards by means of User to Service Latency, criticality of data and high Bandwidth. Such categories are:

- Broadband Enhancement Applications, requiring seamless swapping of Radio Access Technologies (RATs)
- VR/AR (Virtual/Augmented Reality) applications, requiring Low Latency and high Bandwidth
- V2X (Vehicle to Everything) applications that are passing highly critical data, requiring ultra-low latencies

This Thesis will evolve around the latter category, a subset of URLLC (Ultra-Reliable Low-Latency Communication) applications, demanding, per 3GPP (3rd Generation Partnership Project), an ideal latency of as low as 1ms for the User Plane, while other categories mentioned above (such as Broadband Enhancement Applications) have set this requirement to a less restricting range such as 4ms.

Based on open-source software implementations of LTE Systems (OpenAirInterface), functionality is added for a Heterogeneous Disaggregated Setup, providing the ability for a network user to connect via 3GPP (LTE) and non-3GPP (WiFi) technologies. This design is further extended with Multiple-Access Edge Computing Units. These units will provide the Network with operations that are essential for a future 5G MEC-Enabled Vehicular Network. Experimentation and evaluation of the implemented scheme can be carried out in a real Testbed (NITOS Testbed), with USRPs (Universal Software-Defined Radio Peripherals) serving as the network components, down from the end User, up to the Core Network.

Next, we showcase the behaviour of typical mobile user QoS (Quality of Service) data in a visual manner and build around various statistical inferencing tests. A pipeline of operations is then set up for data preprocessing, statistical inferencing, model fitting and finally predicting, both spatially and temporally, enabling proactive decisions in our formed network. This framework can be the basis of a personalized AI solution that can ease on the need of long record-keeping of historic behaviour and is quick to respond to sudden impulses. Worth mentioning is its ability to generate up-to-date & online Radio Environment Maps (REMs), depicting a collaborative users' view of the network to the network itself.

Noted: As of the indexing procedure in this thesis:

- *The dot "." symbol denotes the introduction of a subchapter*
- *".0" denotes the subchapter 0, which is used throughout for chapter-related prerequisite knowledge*

Περίληψη

Χαράσσοντας την διαδρομή μας προς τα επερχόμενα δίκτυα 5ης γενιάς, θα χρειαστεί πρώτα να αναλύσουμε μια νέα γκάμα προβλημάτων που προκύπτουν από την χρήση νέων τεχνολογιών και εφαρμογών. Τα προβλήματα αυτά είναι εμφανή στις ήδη χρησιμοποιούμενες τεχνολογίες και μπορούν να αντιμετωπιστούν μέσω αποτελεσματικής σχεδίασης και υλοποίησης Αρχιτεκτονικής Δικτύου. Έχοντας ως σκοπό την διάχυση και την προσωποποίηση των υπηρεσιών που προσφέρονται σε ένα κινητό χρήστη, είναι σημαντικό να αναβαθμίζουμε συνεχώς το υπάρχον τοπίο από την σκοπιά των υιοθετούμενων τεχνολογιών. Πρακτικά αυτό συμβαίνει μέσω της ενίσχυσης, οικοδόμησης, δοκιμής και ισορρόπησης, έτσι ώστε τα οποιαδήποτε προυπάρχοντα όρια να συναντηθούν και μετέπειτα να επεκταθούν, παράλληλα με την πρόσφατη θέσπιση πρωτοποριακών ιδεών (mmWave, Beamforming, εκτενής διάσπαση και εικονικοποίηση των συστατικών στοιχείων του Δικτύου, κ.λ.π.).

Ένα εκτενές σύνολο από οραματιζόμενες ή σε πρώιμο στάδιο ανάπτυξης εφαρμογές, έχουν διαμορφώσει τόσο σε συλλογικό όσο και σε ατομικό επίπεδο, τις απαιτήσεις τους ως προς την Καθυστέρηση Πρόσβασης Χρήστη στην Υπηρεσία (User to Service Latency), την κρισιμότητα των δεδομένων που πραγματεύονται και το Εύρος Ζώνης (Bandwidth) που χρειάζονται για να λειτουργήσουν. Τέτοιες κατηγορίες εφαρμογών είναι:

- Εφαρμογές Βελτιστοποίησης Ευρυζωνικής Κάλυψης (Broadband Enhancement Applications), οι οποίες δημιουργούν την ανάγκη για την απρόσκοπτη εναλλαγή μεταξύ Τεχνολογιών Πρόσβασης στο Δίκτυο (Radio Access Technologies – RATs)
- Εφαρμογές Εικονικής/Επαυξημένης Πραγματικότητας (Virtual/Augmented Reality Applications), οι οποίες απαιτούν χαμηλούς χρόνους Καθυστέρησης και υψηλό Εύρος Ζώνης
- Εφαρμογές Συνδεδεμένων Αυτοκινήτων (V2X – Vehicle to Everything Applications), οι οποίες πραγματεύονται κρίσιμα δεδομένα, απαιτώντας συγκριτικά τους μικρότερους χρόνους Καθυστέρησης

Η παρούσα διπλωματική εργασία θα εξελιχθεί γύρω από την τελευταία κατηγορία, η οποία αποτελεί υποσύνολο της URLLC κατηγορίας εφαρμογών (Ultra-Reliable Low-Latency Communication Apps). Οι παραχθείσες προαπαιτήσεις αυτής της κατηγορίας, σύμφωνα με τον οργανισμό προτυποποίησης της 3GPP (3rd Generation Partnership Project), διαμορφώνονται στα 1ms ως προς την Βέλτιστη Καθυστέρηση για την μεταφορά Δεδομένων ενός Χρήστη (User Plane), σε σύγκριση με την αντίστοιχη καθυστέρηση 4ms που διαμορφώνεται για τις εφαρμογές Βελτιστοποίησης Ευρυζωνικής Κάλυψης.

Χρησιμοποιώντας υλοποιήσεις LTE συστημάτων ανοιχτού λογισμικού (OpenAirInterface) ως βάση, προστίθεται λειτουργικότητα για την διαμόρφωση μιας Ετερογενούς Διασπασμένης Αρχιτεκτονικής, η οποία επιτρέπει σε έναν Χρήστη να αποκτήσει ταυτόχρονη συνδεσιμότητα στο Δίκτυο μέσω 3GPP (LTE) και μη-3GPP (WiFi) τεχνολογιών. Το σχέδιο αυτό επεκτείνεται κατάλληλα για να φιλοξενήσει μονάδες Υπολογιστικής Πολλαπλής Πρόσβασης στην Άκρη του Δικτύου (Multiple-Access Edge Computing Units). Οι μονάδες αυτές είναι ικανές να προσφέρουν στο Δίκτυο λειτουργικότητες, συνυφασμένες με την ανάπτυξη μελλοντικών 5G Δικτύων Αυτοκινούμενων Οχημάτων, ενεργοποιημένων μέσω μονάδων MEC. Ο πειραματισμός και η

αξιολόγηση του υλοποιημένου συστήματος είναι εφικτά σε ένα ρεαλιστικό περιβάλλον πειραματισμού (NITOS Testbed), όπου Συσκευές Ραδιοεκπομπής καθορισμένες από Λογισμικό (USRPs-Universal Software-Defined Radio Peripherals) υποδύονται τον ρόλο των συστατικών στοιχείων του Δικτύου μας, ξεκινώντας απο τον Τελικό Χρήστη, μέχρι και τον Πυρήνα του Δικτύου.

Έπειτα εκθέτουμε οπτικά την συμπεριφορά τυπικών δεδομένων Ποιότητας Υπηρεσίας (Quality of Service) από χρήστες οι οποίοι χαρακτηρίζονται από κινητικότητα και χτίζουμε γύρω από διάφορα τεστ στατιστικών συμπερασμάτων. Με αυτό τον τρόπο διαμορφώνεται ένα πλαίσιο από διαδικασίες σχετικές με προεπεξεργασία δεδομένων, στατιστικών συμπερασμάτων, προσέγγισης μέσω μαθηματικών μοντέλων και τελικά πρόβλεψης, τόσο ως προς τον χώρο, όσο και ως προς τον χρόνο, ενεργοποιώντας προληπτικές αποφάσεις στο περιβάλλον του Δικτύου μας. Το σύνολο αυτών των διαδικασιών μπορεί να αποτελέσει την βάση για μια προσωποποιημένη λύση Τεχνητής Νοημοσύνης, η οποία μπορεί να ελαφρύνει από την ανάγκη για μακρές καταγραφές ιστορικών συμπεριφορών, όντας ευέλικτη σε ξαφνικές αλλαγές. Αξίζει να σημειωθεί η δυνατότητα να δημιουργούνται συνεχώς ανανεώμενοι Ραδιοπεριβαλλοντικοί Χάρτες (REMs - Radio Environment Maps), συνεργατικά προσφέροντας την εικόνα του Δικτύου από την σκοπιά των χρηστών στο ίδιο το Δίκτυο.

Όσον αφορά την αρίθμηση των ενοτήτων στην παρούσα διπλωματική:

- *Το σύμβολο-τελεία "." ορίζει την έναρξη μίας υποενότητας.*
- *Ο δείκτης ".0" ορίζει την υποενότητα 0, η οποία χρησιμοποιείται για την παρουσίαση προαπαιτούμενης γνώσης για το κεφάλαιο.*

This work is dedicated to the ones who loved and cherished me throughout my course of life as an undergraduate student. I strive to give back to you even half the things you have given me.

For one, this means my parents Panagiotis and Katerina and my brother Alex, being wonderful people who supported me by all means possible.

It also means that I have made quite some friends that I'm very lucky to have by me - you know who you are and I'm proud for each and every one of you.

It would be vague not to explicitly thank my friend and colleague Giorgos Tziokas for his immeasurable open-heartedness, lust for life and timely help.

I would like to thank Prof. Athanasios Korakis for exposing me to interesting research topics at NITLAB, University of Thessaly.

I feel blessed to be working with Nikos Makris, an inspiring doctorate graduate, always providing the needed drive to a curious mind.

Contents

List of Figures

DECLARATION OF AUTHORSHIP

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Ο Δηλών

Χρήστος Νάνης
27/9/2020

Chapter 1 – 5G General

The 5th generation of mobile and wireless communications is expected to greatly impact modern society and industry, aiming beyond Information and Communications Technology (ICT). To begin with, it will greatly leverage peak data rates as compared to previous generations and make room for high user-experienced data rates that are independent of location, in support of enhanced Mobile BroadBand (eMBB) services. It is also expected to enable the next level of human-to-human or human-to-environment interaction, like in the cases of pervasive usage of virtual or augmented reality or free-viewpoint video.

1.1: Introduction to the 5G Concept

5G is expected to clear the path for *Ultra-Reliable Low-Latency Communications* (URLLC) and *massive Machine-Type Communications* (mMTC), landscaping the human-object all-connected world. This will play a catalyst role for developments in vertical industries that will benefit from omnipresent mobile and wireless connectivity. 5G is expected to foster the Industrial Internet, also called as *Industry 4.0*, by permitting reliable and latency-critical communication amongst machines, humans and industrial environments. The automotive and transportation sectors will be revolutionized by advanced forms of collaborative driving, protection of vulnerable road users and leveraged efficiency in railroad transportation. The health sector will experience a new breeze of innovation, where *remote surgery* with haptic feedback and wirelessly operating *smart pharmaceuticals* will be available. Envisioned solutions for *Smart Cities* will be actualized and accelerated, improving quality of life through energy and waste management [1].

Previous generations of communications, focused on one business ecosystem such as mobile broadband in the case of LTE, whilst having a monolithic system design. This generation targets from the early research stages on multi-service and multi-tenancy support and is to comprise a variety of tightly integrated radio technologies , like enhanced LTE (eLTE) , Wi-Fi and 5G radio interfaces , tailored to different frequency bands , cell sizes and service needs [1].

1.2: 5G Service Types Overview

The critical 5G Service types are articulated around the following [1]:

- **Enhanced Mobile Broadband (eMBB):** Related to human-centric applications and enhanced access to multimedia content, services and data with leveraged performance and seamless UX. This Service type, seen as an evolution of currently available LTE services, covers Use Cases with different demands by means of user density, traffic capacity and user mobility but has a common need for seamless radio coverage that is irrelevant to location and timing.
- **Ultra-Reliable and Low-Latency Communications (URLLC):** Fundamental for Industry 4.0, related to Use Cases with strict requirements on latency, reliability and availability. Examples include but are not limited to, remote medical surgery, distribution automation in a smart grid, transportation safety, V2X.
- **massive Machine-Type Communications (mMTC)**: Characterized by a large number of connected low-cost and long battery-time devices transmitting low volume of non-delay-sensitive data, related to agricultural applications, smart-metering and logistics applications.

1.3: 5G Key Performance Indicators (KPIs)

A set of defined parameters as key Network capabilities by IMT-2020[1]:

- **Peak data rates**: Maximum achievable data rates, under ideal conditions per user or device in bps
  - 20 Gbps for Downlink (DL) and 10 Gbps for Uplink (UL) traffic
- **Peak Spectral efficiency**: Maximum data rate under ideal radio conditions, normalized by the channel bandwidth (measured in bps/Hz)
  - 30 bps/Hz for DL and 15 bps/Hz for UL. This combined with the above peak data rates, result in the need for a 2-3 GHz spectrum
- **User experienced data rate**: Achievable data rate available ubiquitously across the coverage area to a mobile user or device in bps
  - 100 Mbps in the DL and 50 Mbps in the UL. This KPI corresponds to the 5% point of the CDF (Cumulative Density Function) of the user throughput
- **5th percentile user spectral efficiency**: Referring to the 5% point of the CDF of the user throughput, normalized by the channel bandwidth in bps/Hz.
  Minimum requirements per scenario are:
  - 0.3 bps/Hz for DL & 0.21 bps/Hz for UL in cases of Indoor Hotspot areas
  - 0.225 bps/Hz for DL & 0.15 bps/Hz for UL in cases of Dense Urban areas
  - 0.12 bps/Hz for DL & 0.045 bps/Hz for UL in Rural areas
- **Average spectral/spectrum efficiency**: Average data throughput per unit of spectrum resource and per cell in bps/Hz/cell
  - 9 bps/Hz/cell for DL & 6.75 bps/Hz/cell for UL in cases of Indoor Hotspot areas

  - 7.8 bps/Hz/cell for DL & 5.4 bps/Hz/cell for UL in cases of Dense Urban areas
  - 3.3 bps/Hz/cell for DL & 1.6 bps/Hz/cell for UL in Rural areas
- **Area Traffic Capacity**: Total traffic throughput served per geographic area in Mbps/square meter
  - 10 Mbps / square meter for the DL in Indoor Hotspot areas
- **User Plane Latency**: RTT of the Radio Network
  - 4 ms for eMBB services and 1 ms for URLLC services
- **Control Plane Latency**: Transition time from idle to active state
  - Less than 20 ms
- **Connection density**: Total number of connected and/or accessible devices per unit area
  - 1M devices / square kilometer for mMTC services
- **Energy Efficiency**: On the network side, referring to the quantity of information bits transmitted to or received from users, per unit of energy consumption of the Access Network. On the device side, to the quantity of information bits per unit of energy consumption of the communication module. Both measured in bits/Joule.
  - Are yet to be set, but support is specified for high sleep ratios and long sleep durations
- **Reliability**: Success probability of transmitting a data packet before a given deadline
  - MAC transmissions of 32 bytes in less than 1 ms in the cell edge of a dense urban test environment as 99.999% probable
- **Mobility**: Maximum speed at which a defined QoS and seamless transfer between radio nodes which may belong to different layers and/or RATs can be achieved.
  - For a rural test environment, the normalized traffic channel link data rate at 500km/h, reflecting the average user spectral efficiency, must be larger than 0.45 bps/Hz in the UL
- **Mobility Interruption time**: The time during which the device cannot exchange data packets because of handover procedures
  - 0 ms, meaning that connections to new cells have to be established first, before dropping the old ones.
- **Bandwidth**: Maximum aggregated system bandwidth
  - At least 100 MHz but with support for over 1 Gbps

Chapter 2 – Evolution of the Access Network Architecture

We can take a step back to reflect on how the network architecture has gradually evolved across each Generation (G), in order to better perceive what innovations have taken place until now and have led to our Base Setup, upon which we continuously build. According to operators, a generation of the network refers to the deployment of a new non-backward-compatible technology. Picking up from where analog circuit-switched 1G networks left off, we continue with an outline of each generation's components, characteristics and utilized technologies.

2.0: Circuit switching vs Packet Switching

Switches are Link-layer devices that stand in-between multiple links, while **switching** is sharing a transmission medium for different hosts in a multi-host network. **Circuit Switching** (CS) and **Packet-Switching** (PS) are two important techniques for data transmission between end points. The former is employed in earlier generations of networks for voice transmission and the latter is used in later generations, either in conjunction with the former or in a standalone manner. Some key differences are:

- CS is *connection-based* and is, subsequently, path inflexible, while PS has a *connectionless* orientation, resulting in its flexibility.
- CS is a Physical Layer technology, while PS is a Transport Layer technology.
- PS employs intelligent collection schemes, by reordering the out-of-order received packets and reassembling any fragmented data, while CS receives in whole-load transmission bursts.
- CS can utilize either *Time Division Switching* or *Space Division Switching*. PS can utilize either *Virtual Circuit Switching* or *Datagram Packet Switching*.

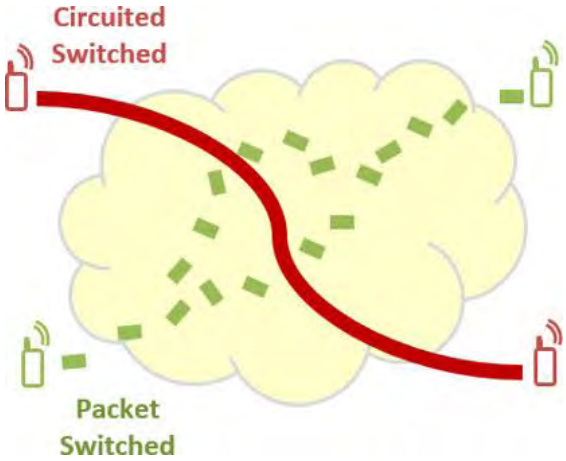

Figure 1: Packet vs Circuit Switching

2.1: On 2G

Second-Generation GSM (2G), introduced in the early 1990's, was the first cellular telephone technology to use digital encryption of conversations. 2G networks were also the first to offer data services and SMS text messaging, while having the lowest data transfer rates compared to their successors. GSM networks employ CS techniques, in contrast to their generational successors of PS or mixed techniques.
GSM architecture is divided in 4 compartments:

- **Mobile Base Station** (MS)
- **Base Station Subsystem** (BSS) - Radio Path Control
- **Network & Switching Subsystem** (NSS) - Call Control
- **Operation and Support Subsystem** (OSS)

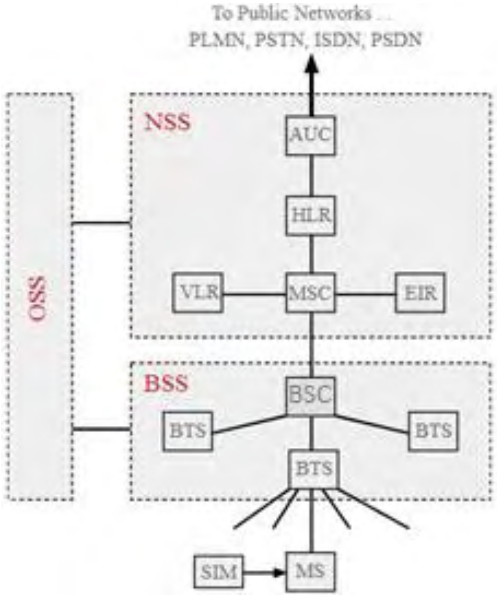

Figure 2: GSM Network Architecture Schematic

The ***OSS*** component is connected to components of the NSS and BSS sections and is used for GSM control and monitoring, while also utilized to regulate the traffic load to the BSS. As the number of Base Stations increases, to cope with scalability needs due to more connected users, some functionality can be transferred to the BTS, cutting back on operation costs. As of the ***MS*** component, it provides the receptor for SMS messages, enabling a user to interchangeably transmit voice and data. It also facilitates access to voice messaging schemes, amongst several data services offered by a GSM network, divided in *telephony*, *bearer* and *supplementary* types.

The technologies used are either TDMA (Time Division Multiple Access), which divides signals into different time slots or CDMA (Code Division Multiple Access), which allocates a special code to each user in order to communicate over a multiplex physical channel. CDMA offers the benefits of multipath diversity and soft handoffs.

Additional key architecture parts are noted:

- **Mobile Services Switching Center** (MSC) is a critical component of the Network & Switching Subsystem. Responsible for switching between PSTN and ISDN, while registration, call location, authentication, routing calls and inter-MSC handovers are some of its responsibilities.
- **Base Transceiver Station** (BTS) of the BSS possesses radio transmitter-receivers and associated antenna to communicate directly with mobile users. Signal quality and radio signal levels are monitored and reported to the BSC.

GSM deployments varied from 900 MHz to 1900 MHz in terms of full channel bandwidth. The latter proved beneficial for a larger group of connected users, especially for the Dense Urban areas case. Speeds varied from initially 9.6 Kbit/s to later on 14.4 Kbit/s.

2.2: On 3G

Third-Generation UMTS (3G) is a successor of a series of improvements and modifications to the "vanilla" GSM, originating from what can be found as 2.5G, which incorporated new techniques such as the General Packet Radio Services (GPRS). Packet-switching (PS) brought new flexibility to the network with demand-based resource allocation and permitting mobility of users. UMTS is an acronym for Universal Mobile Telecommunications System, based on its GSM predecessor and operating at 2GHz with a 5MHz allocated channel bandwidth.

UMTS offers services like speech, SMS and bearer services which provide the capability for information transfer between access points. Negotiating/renegotiating the characteristics of a bearer service at session or connection establishment and/or during an ongoing connection is possible. Provided capabilities also include oriented and connectionless services for Point to Point and Point to Multipoint communication.

The aforementioned services have QoS parameters which vary, in terms of maximum transfer delay, delay variation and bit error rate. Data rate targets are:

- 144 Kbits/s satellite and rural outdoor.
- 384 Kbits/s urban outdoor.
- 2048 Kbits/s indoor and low range outdoor.

There are also QoS classes defined for the offered UMTS Network Services:

- Conversational class (voice, video telephony, video gaming)
- Streaming class (multimedia , video on demand, webcast)
- Interactive class (web browsing , network gaming , database access)
- Background class (email , SMS ,downloading)

The technology used is WCDMA (Wideband Code Division Multiple Access), conceived by 3G Partnership Program (3GPP). It features a Frequency Division Duplex (FDD) Mode which employs both codes and frequencies for separating Users. One frequency is used for the downlink (DL) case and one for the uplink (UL) case. The other mode is the one of Time Division Duplex (TDD), which employs codes, frequencies and time for separating users, while the DL and UL used frequency is the same. As an air interface technology, WCDMA is able to greatly leverage a signal's bandwidth. UMTS architecture is comprised of the following entities:

- **Core Network** (CN) – based on GSM network with GPRS
- **UMTS Terrestrial Radio Access Network** (UTRAN) – air interface access method for a UE
- **User Equipment** (UE)

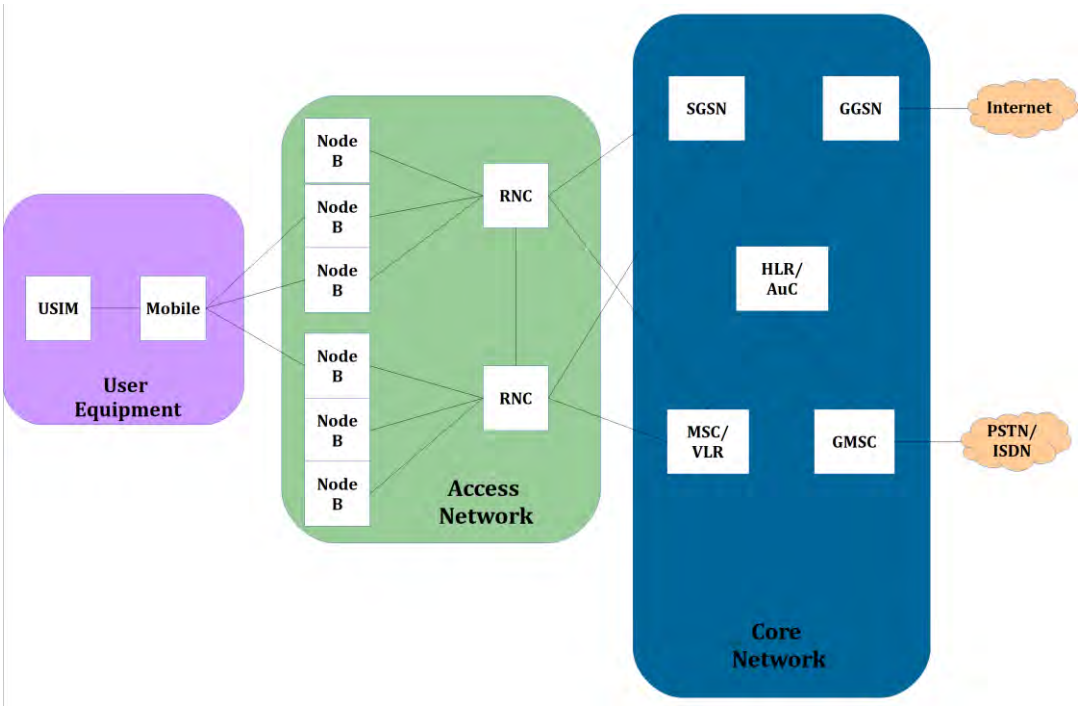

Figure 3: UMTS Network Architecture Schematic

In this setup [2], Base Stations are referred to as *Node-Bs* and the control equipment for a Base Station is called a *Radio Network Controller* (RNC). All are compartments of the **Radio Access Network** (RAN) part, where:

Functions of Node-Bs include:

- Air interface transmission/reception
- Modulation/demodulation
- CDMA Physical Channel coding
- Error handling

Functions of RNC include:

- Radio Resource Control (RRC)
- Admission control
- Channel allocation
- Power control settings
- Handover control
- Ciphering
- Segmentation/reassembly and broadcast signaling

Noted also is the necessity for the network to know the approximate position of the UE so as it can perform its paging/scheduling capabilities. System areas can be:

- UMTS systems (including satellite)
- Public Land Mobile Network (PLMN)
- MSC/VLR or SGSN
- Location Area

- Routing Area (PS domain)
- UTRAN Registration Area
- Cell

The **Core Network** part consists of both CS and PS domains. Circuit-switched domain examples are Mobile services Switching Center (MSC), Visitor Location Register (VLR) and Gateway MSC. Packet-switched domain examples are Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Elements like EIR (Equipment Identity Registry), HLR (Home Location Registry), etc. are shared by both domains. UMTS core transmission utilizes Asynchronous Transfer Mode (ATM). Core Network architecture is subject to modifications upon the arrival of new implementations of services and features. For example, Number Portability DataBase (NPDB) is used to enable a user to change networks while keeping his/her own phone number.

As of the **User Equipment**, the UMTS standard is non-restrictive on the UE functionality in any way, while terminals work as air interface counterparts for Node-Bs with various types of identities, mostly taken directly from GSM specifications:

- International Mobile Subscriber Identity (IMSI)
- Temporary Mobile Subscriber Identity (TMSI)
- Packet Temporary Mobile Subscriber Identity (P-TMSI)
- Temporary Logical Link Identity (TLLI)
- Mobile Station ISDN (MS-ISDN)
- International Mobile Station Equipment Identity (IMEI)
- International Mobile Station Equipment Identity and Software Version (IMEI-SV)

UMTS mobile station can operate in one out of 3 modes available:

- PS/CS mode: Mobile Station is attached to both the Packet-Switched and Circuit-Switched domain and is capable of simultaneously operating Services of either domain.
- PS mode: Mobile Station is attached to the Packet-Switched domain only and may only operate services of this domain. This does not, however, prevent CS-domain services to be offered over PS-domain services (especially in the case of VoIP).
- CS mode: MS is attached to the Circuit-Switched domain only and may only operate services of the respective domain.

A UMTS IC card is the physical equivalent of a GSM SIM card.

- Supports one User Identity Module (USIM) application or, optionally, more than one.
- Supports one or more user profiles on the USIM
- Supports over-the-air information update
- Security and user authentication
- Optional inclusion of payment methods
- Optional secure downloading of new applications.

As a note , Third Generation UMTS did not live up to the initial expectations , not being able to offer a genuine next-generation experience to the user and left a lot to be desired, a gap that was later filled by Fourth-Generation Networks.

2.3: On 4G

Market-formed needs of Network Services, wireless Content Distribution and access to large amounts of information, surfaced the need for development of Broadband Telecommunication Systems and for the successor, **Fourth Generation (4G)** Networks.
Long-Term Evolution/System Architecture Evolution (LTE/SAE) marketed as 4G, specified by *3rd Generation Partnership Project* (3GPP), is a mobile generation which supersedes the 3.75G (HSPA+) and 3G (UMTS) families of standards , with the aim of achieving high system capacity, leveraged peak data rates and low latency, all while having reduced operating costs, support for MIMO and flexible bandwidth operations.

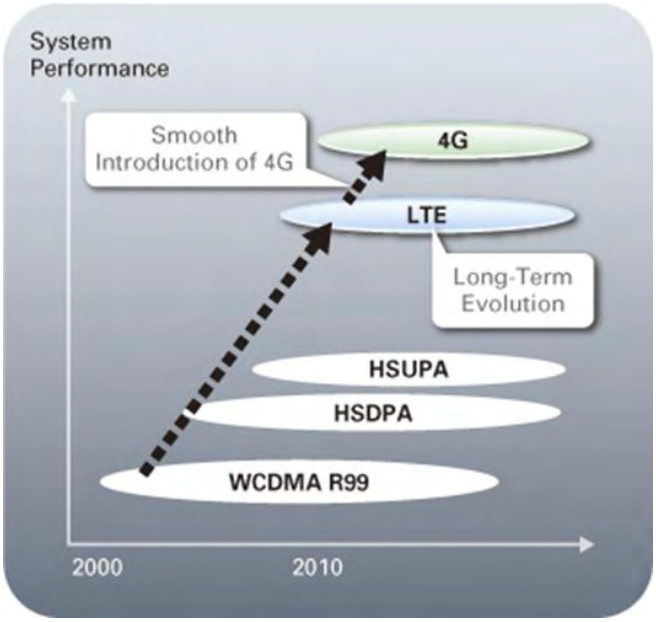

Figure 4: Generations of networks across the years

4G systems do not support circuit-switched telephony service domains and rely exclusively on all-Internet Protocol (all-IP) communication, as in the case of IP telephony. Users are offered services without constraints, in time and location, enhanced connectivity to the Access Network with guaranteed QoS and low latency. There is also support for new generation IP Protocols (IPv6) and Multicasting, especially useful for electronic market ventures. The peak data rates are formed in the range of 75 Mbps in the UL case and 300 Mbps in the DL case, while connectivity to WLAN, satellite and legacy systems (GSM) is feasible, following a Heterogeneous paradigm. Used frequencies are higher (up to 5GHz) with an allocated spectrum of 20 to 100 MHz per channel.

Key features observed in 4G-labelled technologies are:

- **Multiple-Input-Multiple-Output** (MIMO): MIMO Operations perform spatial multiplexing while utilizing pre-coding and transmission diversity. Such operations tackle efficiently problems arising from reflections of signals, opposed to previous generation networks. Peak data rates are further leveraged by additional signal paths that are created by a number of dissimilar antennas and dissimilar data streams, such as 2x2 or 4x4 schemes.[3]
- **Orthogonal Frequency Division Multiplexing** (OFDM): Technology which provides high resilience to interference and reflections at the same time. Access Schemes are further divided into two access methods used in the DL and UL (Downlink/Download & Uplink/Upload) cases. In the former case, OFDM multi-carrier is utilized and in the latter, SC-FDMA (*Single-Carrier Frequency Division Multiplexing Access*) is used. The aforementioned access schemes promise a smaller PAPR (*Peak to Average Power Ratio*) and leveraged spectral-efficiencies.
- **Turbo Error-Correcting Codes** for being able to operate at minimum-required received Signal to Noise Ratio (SNR).
- **Adaptive Schemes**, meaning changing the available transmission settings (modulation, rate, coding etc.) depending on the observed radio conditions.

2.3.1: LTE Architecture

LTE architecture is comprised of the following entities:

- **Evolved Radio Access Network** (E-UTRAN)
- **Serving Gateway** (SGW)
- **Mobile Management Entity** (MME)
- **Packet Data Network Gateway** (PDN P-GW)
- **Home Subscriber Server** (HSS)
- **Policy Control and Charging Rules Function** (PCRF)

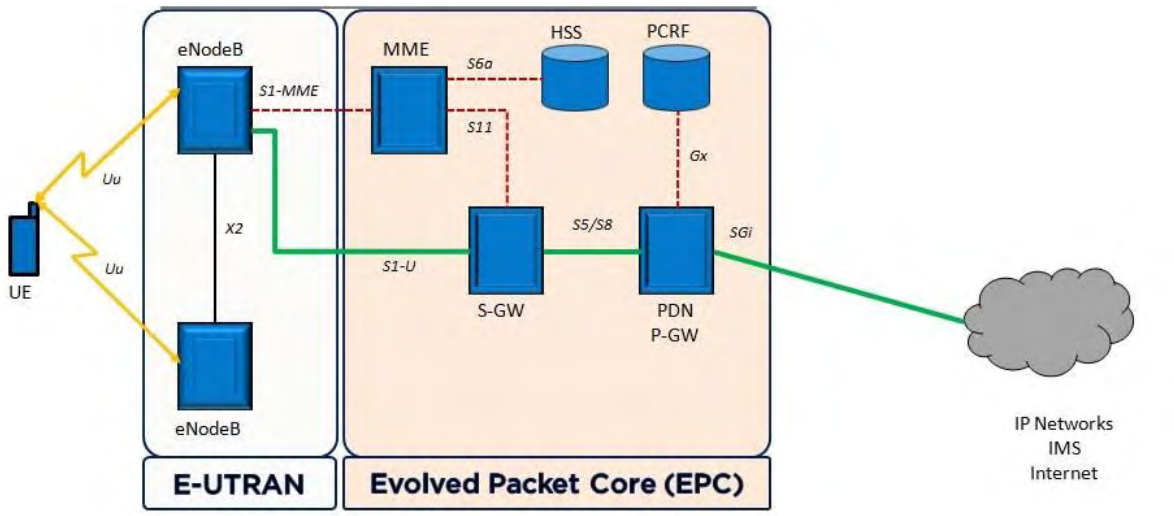

Figure 5: LTE Architecture Schematic

As can be observed in *Figure 5*, it is an interconnected set of base stations i.e. *evolved NodeBs* (*eNBs*), generating a rather flat architecture. In contrast to older-generation deployments, a centralized controller is now absent and the base stations are interconnected via *X2* Interface, enabling them to perform various functionalities e.g. handovers. Cells are connected with the *Evolved Packet Core* (EPC) via *S1* Interface (specifically, S1-U interface is for the eNB - S-GW communication and S1-MME interface is for the eNB - MME communication).

E-UTRAN primarily constitutes a single base station, that is coined the term eNB (evolved NodeB), as noted before. Except providing interfaces for the UE, it hosts a Stack of Protocols that are divided in Layers, denoted as Layer 1/2/3 or L1/L2/L3, respectively. E-UTRAN functions include admission control, radio resource management, scheduling and enforcement of negotiated UL QoS and compression/decompression of DL/UL user plane packet headers.

SGW works as a mobility anchor between LTE and other 3GPP user plane technologies during *inter-eNB handovers* (when a UE moves between cells managed by different eNBs), while directing and forwarding user data. Also manages and/or stores UE context information, such as network internal routing information and parameters of the IP bearer service. It generates paging requests, when DL data for a UE make their arrival, while being responsible for terminating the DL traffic flow to the UE.

MME is a key control node for the Access Network, tracking and paging idle-mode UEs. Responsible for Bearer activation/deactivation procedures and for choosing a dedicated SGW to a specific UE, in the cases of initial attachment and *intra-eNB handovers* (When a UE moves between cells managed by the same eNB). Functionalities also include authenticating a UE via HSS communication, terminating all Non-Access-Stratum (NAS) signaling and generating/allocating temporary UE identities. Also noted, is the mapping of a UE to a Public Land Mobile Network (PLMN), while enforcing UE roaming restrictions, if there are any. Being a termination point of ciphering and integrity protection for NAS signaling, MME finally provides the control plane function for mobility between LTE and legacy networks that utilize the S3 interface.

PDN GW is the gateway which terminates the *SGi* interface. SGi actualizes the connection of a PDN GW to a Packet Data Network (PDN), which can be an external public/private, operator-specific/intra-operator PDN. If a UE has access to multiple PDNs, there may be more than one PGWs for that UE. It acts as a mobility anchor between 3GPP and non-3GPP technologies and provides connectivity from the UE to external PDNs, by being the point of entry or exit of traffic for the UE. It manages policy enforcement, deep packet inspection for user packets, charging support and DHCPv4/v6 functions, amongst other functionalities.

HSS is a commonplace Core Network component, borrowing from the GSM and UMTS eras and its main utilization is for subscriber-related record-keeping, merely as a Database.

PCRF component can actualize policy and pricing schemes and control the flow-based charging duties of the Policy Control Enforcement Function (PCEF) that is collocated with the P-GW.

2.3.2: LTE Protocol Stack & Layered Approach

E-UTRAN is comprised of a Protocol Stack, as mentioned above, that is layered and can be split into subsequences that carry out *Control Plane* (signaling related to user state and configuration) and *User Plane* (user-related data traffic) functionalities. Inter-layer communication is formulated with the use of the terms *Service Data Unit* (SDU) and *Protocol Data Unit* (PDU). Typically, the former denotes input data to a Layer, while the latter denotes output data from a Layer (as input to a lower Layer). Transfer of these Units is actualized by the use of LTE Communication Channels, to be covered in the next section. A general outline is as below:

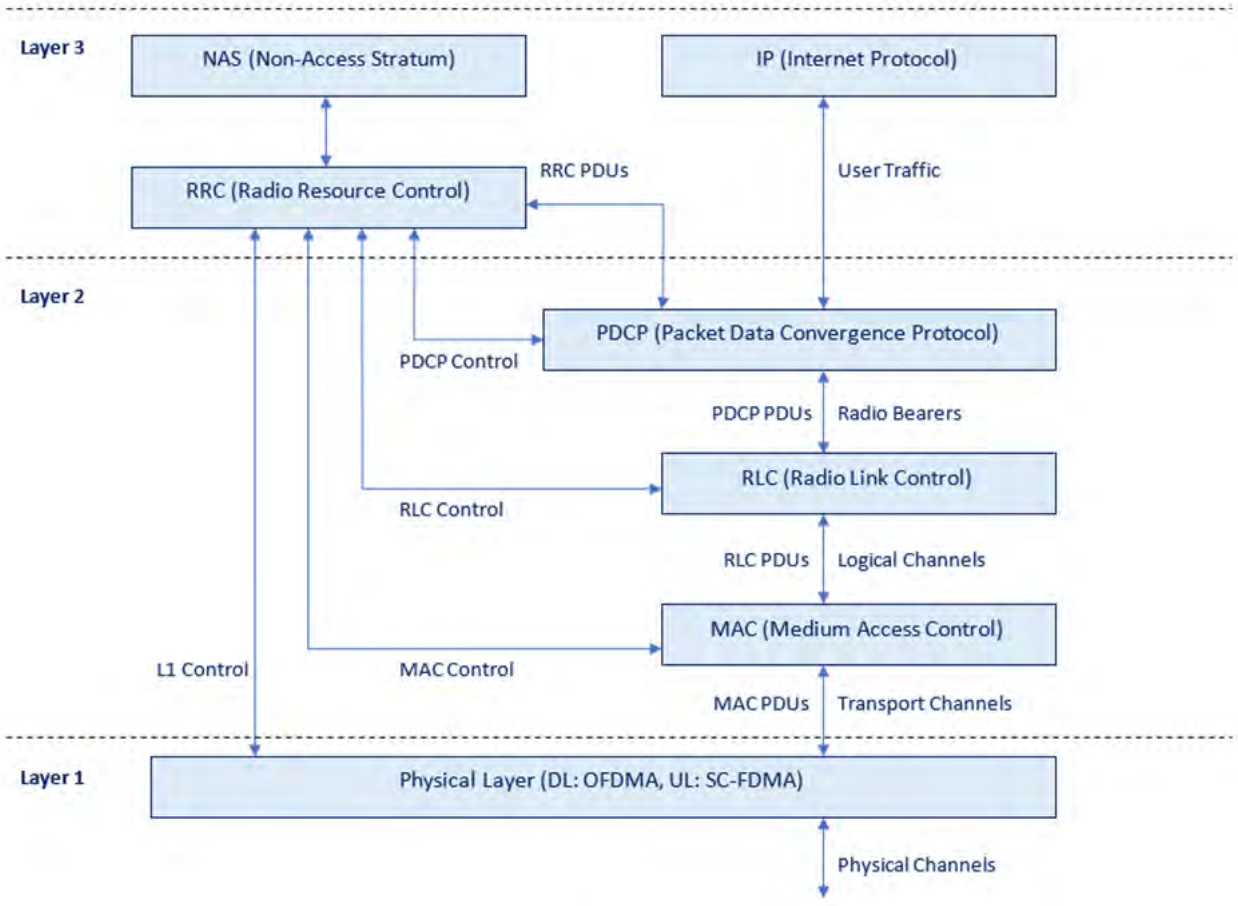

Figure 6: E-UTRAN Protocol Stack Schematic

- **L1**: *Layer 1* is comprised only of the *Physical Layer*, which carries all necessary information from the MAC Transport Channels over the air interface [4]. Responsible for link adaptation, power control, cell search (for initial synchronization and handover procedures) and other measurements for the RRC Layer.

- **L2**: *Layer 2* is comprised of:
    1. *Packet Data Convergence Protocol* (PDCP) Layer
    2. *Radio Link Control* (RLC) Layer
    3. *Medium Access Control* (MAC) Layer

- **L3:** Layer 3 is comprised of:
    1. *Non-Access-Stratum* (NAS) Layer
    2. *Radio Resource Control* (RRC) Layer
    3. *Internet Protocol* (IP) Layer

NAS supports the mobility of the UE and the session management procedures, to establish and maintain IP connectivity between the UE and a PDN GW.

RRC contains services and functions including:
1. Broadcasting of system information related to the *Access Stratum* (AS)
2. Paging, establishment, maintenance and release of an RRC connection between the UE and E-UTRAN
3. Security functions including key management, establishment, configuration, maintenance and release of Radio Bearers

PDCP Layer employs functionalities for:
1. IP header compression and decompression
2. Transfer of Control or User Plane data
3. Maintenance of PDCP Sequence Numbers (SNs)
4. In-sequence delivery of upper layer PDUs at re-establishment of lower layers
5. Duplicates detection and elimination of lower layer SDUs at re-establishment of lower layers for RBs mapped on RLC AM
6. Ciphering and deciphering of User and Control Plane data
7. Integrity protection and verification of Control Plane data
8. Timer-based discarding , duplicate discarding
9. Mapping SRBs (*Signal Radio Bearers*) and DRBs (*Data Radio Bearers*) on DCCH and DTCH Logical Channels

RLC Layer operates in 3 modes: *Transparent Mode* (TM), *Unacknowledged Mode* (UM) and *Acknowledged Mode* (AM). Its functionalities are:

1. Transferring upper layer PDUs
2. Error-correcting through ARQ (AM mode only)
3. Concatenating, segmenting and reassembling RLC SDUs (UM&AM modes)
4. Re-segmenting RLC data PDUs (AM mode)
5. Reordering RLC data PDUs (UM & AM modes)
6. Duplicate detecting and RLC SDUs discarding (UM & AM modes)
7. RLC connection re-establishment
8. Protocol Error Detection (AM mode)

MAC Layer is accountable for:

1. Mapping between Logical and Transport Channels
2. Multiplexing of MAC SDUs , originating from Logical Channels, into a single Transport Block (TB), that will be delivered to the Physical Layer via Transport Channels
3. Demultiplexing of MAC PDUs carried by one or more Logical Channels in TB form that were delivered from the Physical Layer on Transport Channels.
   Note: If there is a one-to-one correspondence between a MAC SDU and a MAC PDU, the size of the MAC SDU can be known implicitly from the TB size. In these instances, a headerless MAC PDU format is used as a transparent MAC PDU [5].
4. Scheduling Information reporting
5. Error Correction through HARQ
6. Priority handling between UEs by means of dynamic scheduling
7. Priority handling between Logical Channels of one UE
8. Logical Channel prioritization

2.3.3: LTE Communication Channels

LTE Layer 2 structure consists, per 3GPP, of the PDCP, RLC and MAC Layers. *Information flows* amongst Protocol Layers are what we call as **Channels**. Transport Channels are located between the MAC and PHY Layers, while MAC multiplexes RLC links and manages scheduling and priority handling via Logical Channels. These channels provide interfaces to each Layer within the LTE Protocol Stack and induce a well-defined and in-order segregation of data. LTE utilizes different types of Logical, Transport and Physical Channels, distinguishable by the information type that is passed through [6].

- **Logical Channels** define the type of information that is aired. Logical Channels are defined for data and signaling information passed between RLC and MAC Protocol Layers
- **Transport Channels** define transmission parameters e.g. encoding used, for data and signaling between MAC and PHY layers
- **Physical Channels** utilize the RF data transmission itself

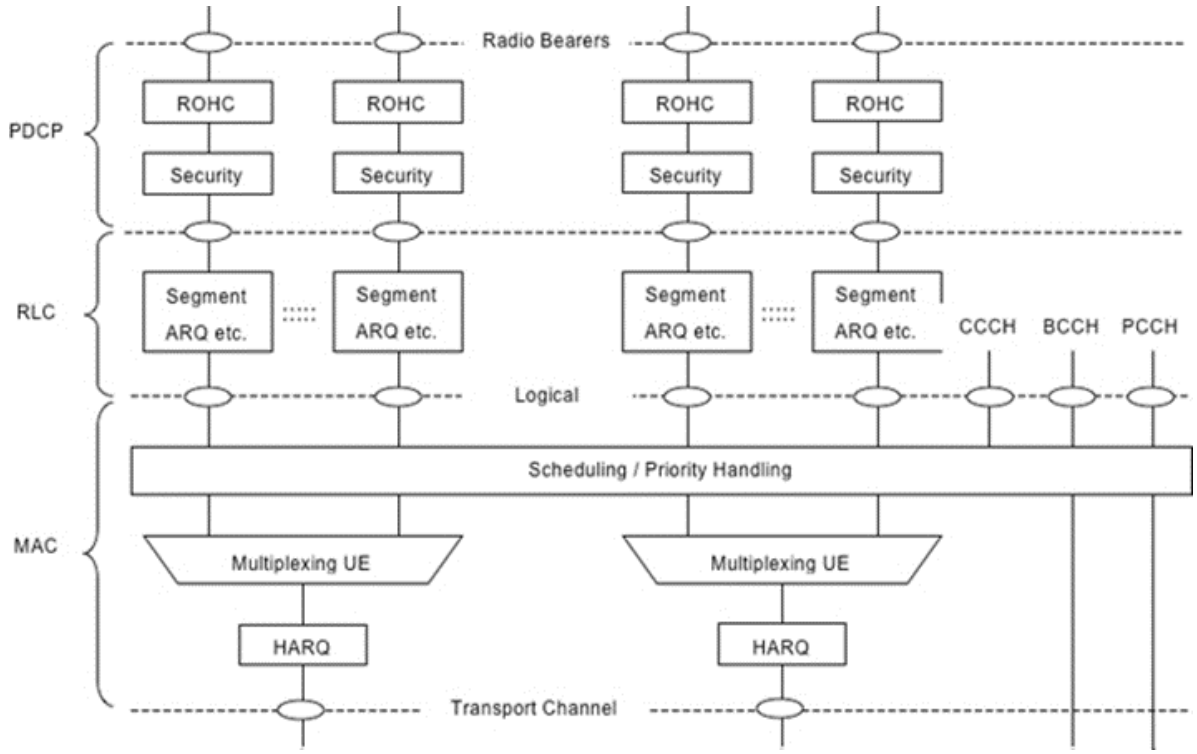

Figure 7: LTE Layer 2 Downlink Structure

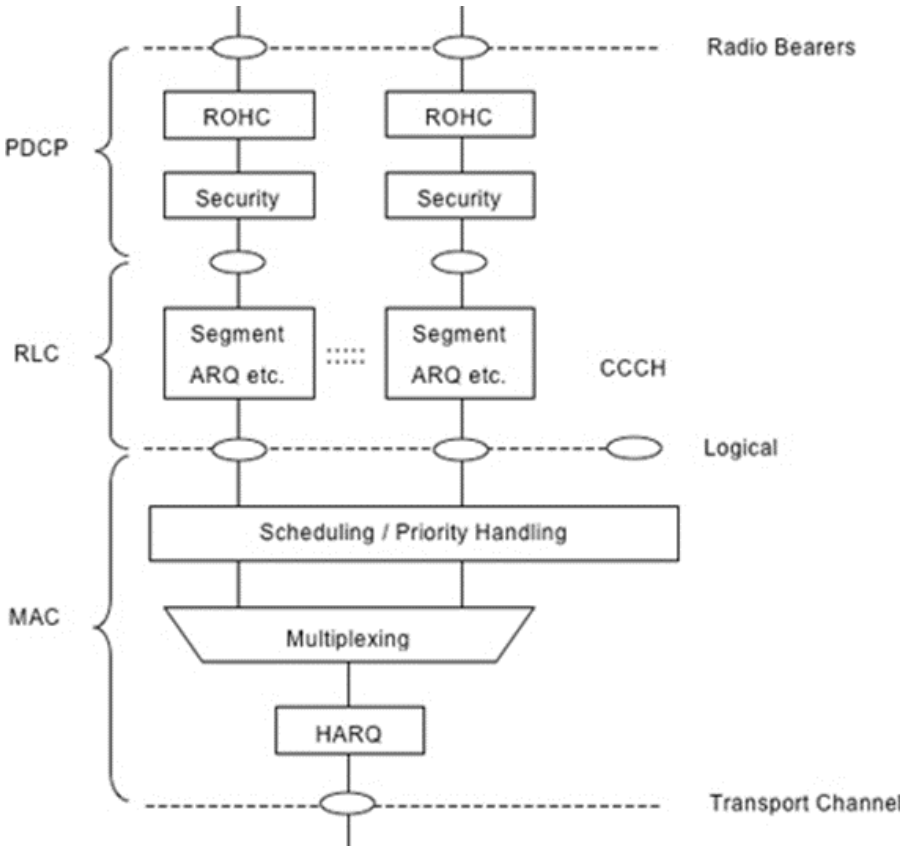

Figure 8: LTE Layer 2 Uplink Structure

2.3.3.1: LTE Logical Channels

There are 2 types of Layer 2 Logical Channels [7], where each logical channel type is defined by the type of information transferred:

- **Control Channels**: For transferring Control Plane information
- **Traffic Channels**: For transferring User Plane information

Control Channels are:

- *Broadcast Control Channel* (BCCH): DL channel for broadcasting system control information
- *Paging Control Channel* (PCCH): DL channel for paging and system information change notifications. This channel is used when the network does not know the location cell of the UE
- *Common Control Channel* (CCCH): For transmitting control information between UEs and network. Used for UEs with no RRC connection with the network
- *Dedicated Control Channel* (DCCH): A point-to-point bi-directional channel, transmitting dedicated control information between a UE and a network. Used by UEs with an RRC connection

Traffic Channels are:

- *Dedicated Traffic Channel* (DTCH): Point-to-point channel, dedicated to one UE for transferring user information. A DTCH can exist in both UL and DL
- *Multicast Traffic Channel* (MTCH): Point-to-multipoint DL channel for transmitting traffic data from the network to the UE. This channel is used only by UEs that receive MBMS (Multimedia Broadcast Multicast Service)

2.3.3.2: LTE Transport Channels

DL Transport Channels [8] are:

- *Broadcast Channel* (BCH):
  1. Has fixed, pre-defined transport format
  2. Imposes the requirement for transmissions to be broadcast in the entire coverage of the cell
- *Downlink Shared Channel* (DL-SCH):
  1. Provides support for HARQ (Hybrid Automatic Repeat reQuest)
  2. Supports dynamic link adaptation by means of varying the modulation, coding and transmit power
  3. Possibility to broadcast in the entire cell and to use beamforming
  4. Provides support for both dynamic and semi-static resource allocation
  5. Provides support for UE discontinuous reception (DRX) to enable UE power saving
- *Paging Channel* (PCH):
  1. Support for UE DRX (DRX cycle is indicated by the network to the UE)
  2. Requirement to be broadcast in the entire coverage area of the cell
  3. Mapped to physical resources which can be used dynamically for traffic or control channels
- *Multicast Channel* (MCH):
  1. Requirement to be broadcast in the entire coverage area of the cell
  2. Supports MBSFN (Multicast-Broadcast Single-Frequency Network) combining of MBMS transmission on multiple cells
  3. Supports semi-static resource allocation (e.g. with a time frame of a long cyclic prefix)

UL Transport channels are:

- *Uplink Shared Channel* (UL-SCH):
  1. Possibility to use beamforming
  2. Support for dynamic link adaptation by varying the transmit power and potentially modulation and coding
  3. Support for HARQ and both dynamic and semi-static resource allocation.
- *Random Access Channel* (RACH):
  1. Limited Control Information
  2. Collision Risk

2.3.3.3: LTE Physical Channels & Signals

DL Physical Channels carry L2 information, while DL Physical Signals (also called as Reference Signals) are only used by the PHY Layer.
Mentioned DL Physical Channels are:

- *Physical Downlink Shared Channel* (PDSCH): Carries the DL-SCH and PCH Transport Channels, with the former containing real user data
- *Physical Downlink Control Channel* (PDCCH): Informs the UE about the resource allocation of PCH and DL-SCH, along with HARQ info related to DL-SCH. It carries the uplink scheduling grant
- *Physical HARQ Indicator Channel* (PHICH) : Carries ACK/NACKs in response to UL transmissions
- *Physical Control Format Indicator Channel* (PCFICH):
  1. Informs the UE about the number of OFDM symbols used for the PDCCHs
  2. Transmissions at each subframe
- *Physical Broadcast Channel* (PBCH): The coded BCH transport block is mapped to four subframes within a 40 ms interval

DL Physical Signals are:

- *Reference Signal*
- *Primary & Secondary Synchronization Signals* (P-SS and S-SS)

UL Physical Channels carry L2 information. Such channels are:

- *Physical Uplink Shared Channel* (PUSCH): Carries the UL-SCH (contains actual user data), ACK/NACKs (Acknowledgments/Non-Acknowledgements), CQI (Channel Quality Indicator)
- *Physical Uplink Control Channel* (PUCCH): Carries ACK/NACKs, in response to DL transmissions. Carries CQI report and SR (Scheduling Request)
- *Physical Random Access Channel* (PRACH): Carries random access preamble

UL Physical Signals are:

- *Demodulation Signal* (DM RS): Associated with transmission of PUSCH and PUCCH, it enables channel estimation and coherent demodulation.
- *Sounding Reference Signal* (SRS): Generated by a UE in the UL direction and utilized by the eNodeB for UL Channel Quality and timing estimation/alignment. Frequency-selective UL scheduling, power control, antenna selection and DL Beamforming are also facilitated by this signal [9].

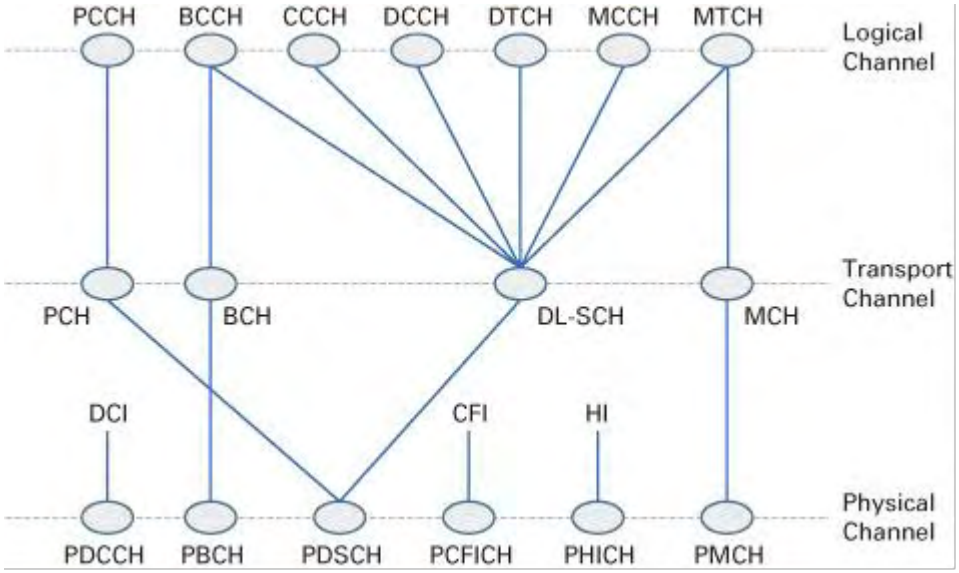

Figure 9: Downlink Channels Schematic

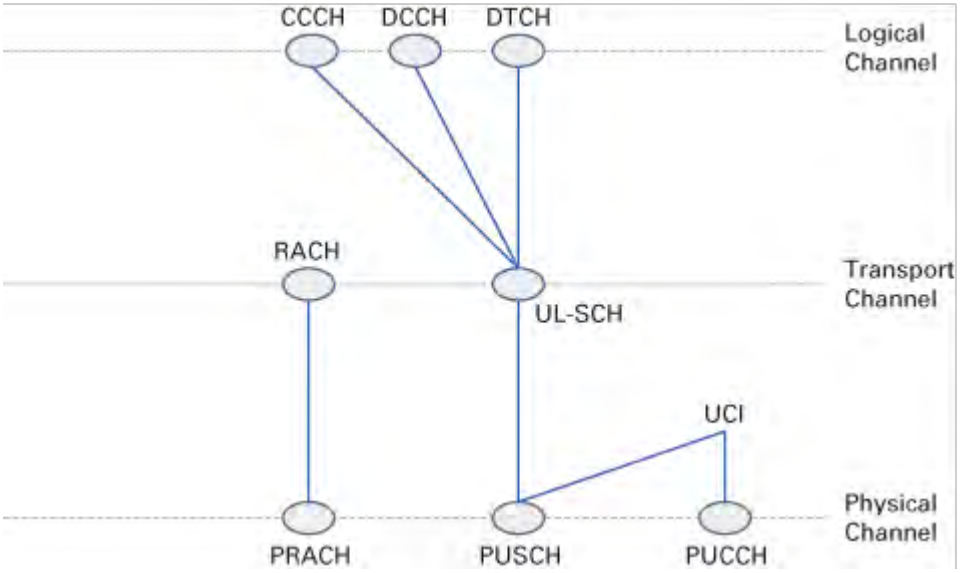

Figure 10: Uplink Channels Schematic

2.3.4: Importance of scheduling in time-critical context

Scheduling is vital for achieving fast-adjusted and efficiently-utilized radio resource allocation, all while transmission times are upper-bounded by restrictive values, as in the case of LTE, set to 1ms. *Transmission Time Interval* (TTI) is defined as the length of time required to transmit a TB. At each TTI, a Base Station Scheduler typically proceeds with:

- Considering the Physical Radio Environment per UE. The UEs report their perceived QoS to the Scheduler, who decides on the Modulation and Coding Scheme.
- Prioritizing the QoS requirements amongst the UEs. Both delay-intolerant real-time services and high peak data rate-demanding communication services are supported.
- Informing UEs on the allocated radio resources. UEs are scheduled both in the DL and UL case. For each UE scheduled in a TTI, the to-be-aired data is carried in a TB, to be delivered on a transport channel. In the DL case, there can be a maximum of 2 generated TBs per TTI per UE, if spatial multiplexing is utilized [10].

2.4: A practical 5G outline

Having outlined the journey that started from legacy to currently-deployed systems, we can introduce some key concepts/enablers and newly-conceived changes in the network architecture that are undoubtedly 5th generation-defining. These changes enable us to set up a wide range of functionalities that are to reside in various future mobile network deployments, leading to variable levels of flexibility and intelligence.

2.4.1: Functional Splits Architecture

Eventually distributing the architecture, aids in speeding up the connection setup and reducing the time required for a handover. An example of how this approach works in our favor, is the LTE MAC Layer, as prior noted for its scheduling responsibilities, that is represented only in the UE and Base Station components. This results in fast communication times and swiftly-made decisions between eNBs and UEs. Originally, in the UMTS era, the MAC protocol and scheduler resided in the Radio Network Controller, whereupon the HSDPA introduction, an additional MAC sub-Layer was introduced in the eNB Protocol Stack.

Operating in the logic of gradually and extensively disaggregating our Protocol Stack and thus our Base Station, as noted in the previous chapter, multiple split options are available e.g. at the PDCP/RLC or the MAC/PHY points, while the ability to be interchangeably used during operation is possible [17].

2.4.1.1: PDCP/RLC functional split

Research work carried out in [11], has shown through real testbed experimentation that the PDCP layer convergence point has lesser requirements than the MAC/PHY one, in terms of extra overhead and restrictions imposed. This split option has evident benefits for utilization, as multiple technologies can be coordinated from a single PDCP/IP instance at the base station, enabling higher network capacity and network selection policies even on a per-packet basis.

Seamless inter-technology mobility for a UE can be achieved when the PDCP/RLC split is used as a convergence sublayer amongst base station components that incorporate technology multi-tenancy in context of 5G Networks. Already implemented and offered by *OpenAirInterface*, following the *New Radio* (NR) specification, it is utilized as standard in the context of this thesis.

The split-defined architecture that is created, consists of 2 new components in our Access Network topology, that collaboratively are the former definition of Base Station eNBs, with technology-agnostic, multi-tenancy support.

- **Central Units** (CUs) incorporate the NAS, RRC and PDCP subset of the Protocol Stack

- **Distributed Units** (DUs) include the RLC and downward layers of the stack.

The created relationship from the CU perspective is 1:N, meaning that a Central Unit can be connected to multiple DUs of either *3GPP* or *non-3GPP* (*WiFi*) technologies, while from the Distributed Unit perspective, the relationship is 1:1, with a DU being associated with only a single CU instance.

2.4.2: X-haul connections and F1 Application Protocol (F1AP)

Some terms are, once more, to be introduced. Disaggregating our system leads to defining the resulted new paths in our base station ecosystem.

- The ***backhaul*** is typically the backbone connection to the Core Network or the Internet. In our context, it's the connections between the CU part of the RAN and the S/P-GW & MME Core components (S1 interface).
- A ***fronthaul/midhaul*** connection is typically the link between the controller or radio head that feeds the next link. In our context, these are the connections between the CU and the multi-technology DUs.

The communication and signaling between these entities is standardized by the *F1 Application Protocol* (F1AP) over the newly introduced F1 interface. The User Plane utilizes the F1-U interface which, in turn, uses GPRS Tunneling Protocol (GTP) encapsulation over UDP/IP links, while the Control Plane utilizes the F1-C interface which uses SCTP/IP associations [12].

In action:

1. A CU first associates via SCTP to the Core Network
2. F1AP protocol is initiated by the DU side, sending to its subsequently registered CU, an F1 Setup Request
3. CU and DUs perform an RRC configuration setup procedure

2.4.3: On Integrating non-3GPP technology

The presented disaggregated setup in *Figure 11* follows a *Heterogeneous* paradigm. Essentially, that means that a UE that connects to the RAN has multiple interfaces to be able to transmit to/receive from DUs of 3GPP (LTE), NR (5G) and non-3GPP (WiFi) technologies. We will elaborate on the latter case, since it presents some key differences from the others in handling and integrating.

As of the Control Plane, the chain of events implemented is the one presented in the previous section. This, along with appropriate and parameterized initial configuration, for transferring those parameters to the execution environment, results to a successful registration and homing of non-3GPP WiFi DU node(s).

Concerning, though, the Data Plane, the WiFi stack poses significant differences compared to the mobile networking stack by means of supported operations. Protocol-wise, the respective network functions performed in the case of DL/UL are called *Data Requests* and *Data Indications.* For example, an RRC Data Request usually precedes a PDCP Data Request, which is, in turn, a predecessor of an RLC Data Request. Those functions utilize communication channels that are created, while the F1oIP protocol agent instruments an asynchronous inter-layer (PDCP/RLC) communication mechanism. Upon the reception of a Data Request (on the DU side), a series of sub-procedures takes place for traversing the payload to the network UE or sending the data back to the CU. These events include the reception of a Data Request originating from the CU side, deserializing and stripping off its PDCP header, before delivering the payload to the wireless driver running atop of the WiFi DU. As of the UL data flow, the reverse process needs to take place, i.e. encapsulating the data around a PDCP header that is formulated from the respective PDCP/IP instance running on the CU side. The added value is essentially scheduling information for mapping traffic to the Logical, Transport and Physical Communication channels of the network. The aforementioned UL procedure includes packet compression and dedicated processes for sequence number assignment to the packets having as destination the CU. To incorporate context information not available in the WiFi case (e.g. protocol context and Data Bearer ID), a lookup table at the DU side is employed, that maps the IP addresses of the receiving clients to the associated protocol context information extracted from data requests [12]. The above was implemented based on the OpenAirInterface disaggregated platform and tested in a real testbed using Ettus' B210 USRPs. The WiFi configuration utilizes 802.11n channels.

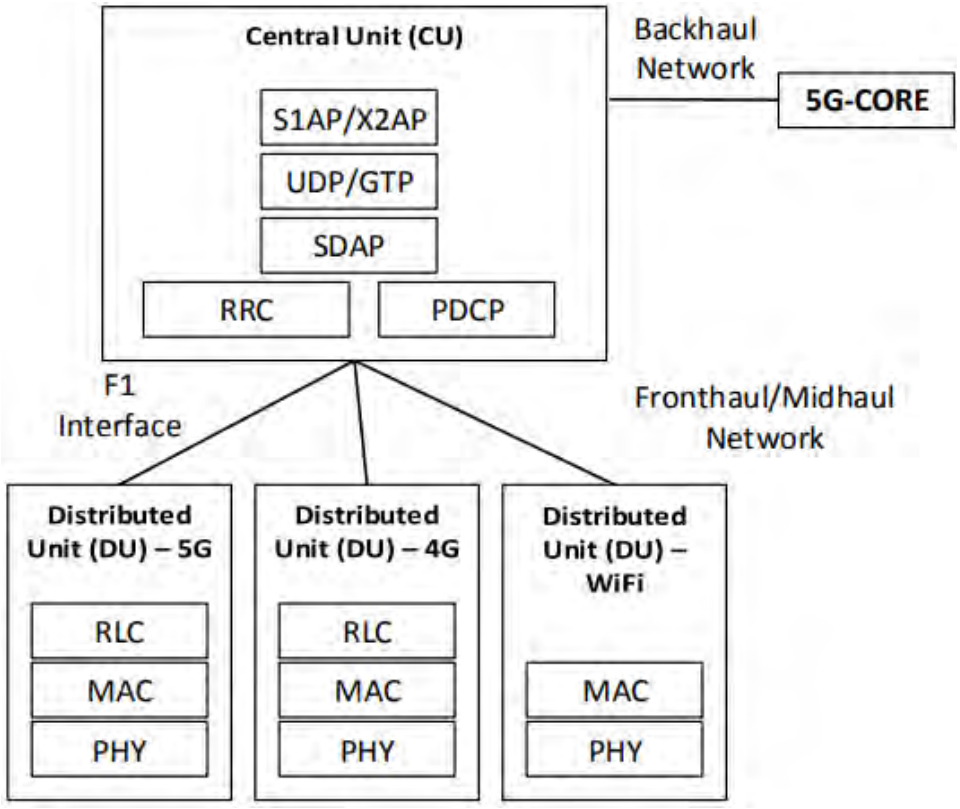

Figure 11: 5G Data Plane Schematic

Chapter 3 – On Multiple-Access Edge Computing

3.0: Introduction to Cloud Computing

Cloud-based mobile applications have skyrocketed in demand through recent years. Applications like real-time image recognition, video streaming, etc., impose restricting requirements for the users' equipment/devices i.e. requiring high data processing capabilities. While most market-distributed smartphones experience constant improvement in various aspects, limitations induced by their size and battery life render them as not appropriate, or at least as a last resort, for executing certain tasks of high complexity. Cloud Computing techniques have, thus, proven to be useful for users, cutting back on energy consumption via offloading to remote services and are generally responsible for carrying out what is dictated by the network users of a demanding application.

Cloud structure is usually centralized, imposing a large geographical gap between the users and the services provided, thus, contributing to an unwantedly high end-to-end communication latency and multiple in-between hops. Ingress Bandwidth to the cloud can also experience saturation, due to the many-to-one User-Cloud relationship [13].

3.1: Introduction to Multiple-Access Edge Computing

The observations above, surfaced a need for installing computing and infrastructural units at the Network's Edge, meaning as close as possible to the End User, addressing directly the Latency and Bandwidth issues. The mentioned units are called **Multiple-Access Edge Computing** units, merely, mobile Edge Clouds, placed 1 or 2 hops away from the user, making Cloud-User communication fast and tackling congestion-related issues to the Backhaul Network. MEC can play a catalyst role in assisting the User-Service communication. Such roles are:

- Providing service content and resources with high availability through selection of technologies
- Decide on future reliability of offered services
- Ability to change physical location, seamlessly, statefully and live, depending on Radio Environment Quality or with respect to the location and mobility of connected users
- Provide energy-efficient and battery life-enhancing transmission schemes.

MEC can be considered as a perfect key enabler for various real-time and context-aware technologies that will enhance interconnectivity and intercommunication, all while strengthening collaborative operation in the high-responsive and intelligent Cloud-RAN. Under the MEC-enabled paradigm, operators will be granted the freedom to employ and collaborate with various authorized third-parties to build upon their RAN edge service ecosystem. Innovative applications can be deployed for mobile subscribers, enterprises and vertical segments. The MEC concept is also found to be further divided into *Network-centric*

(e.g. in the case of local connectivity and caching), *Information-centric* (e.g. in the case of content optimization) and *Device-centric* (e.g. client computational offload) categories [14].

3.2: MEC-deployed Services and the VM vs Container Dilemma

There exist four types of considered MEC-deployed Services [14]:

- **Common Services**: Fundamental part of the MEC Service ecosystem. Will facilitate the usage of real-time network and radio information. As of the control plane, an abstract view of the network status (e.g. topology and connectivity context) can be obtained by extracting the necessary RAN parameters, while having a pre-defined level of granularity. On the data plane, MEC applications will be accessible as IP service endpoints while record-keeping the underlying network status and configuration in a local database scheme.
- **Platform Services**: Physical and/or virtual resource allocation in terms of computation, storage, network and I/O. Additional flexibility can be achieved through Platform Services, allowing an on-cloud service execution in an isolated and tenant-based manner. Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) are mentioned as orchestrators to further allocation intelligence.
- **Support Services**: Common, baseline functionalities to be utilized further by more complex and sophisticated services. A minimal example will include communication modules, discovery and registry utilities, authentication/ authorization, policy/charging schemes and service-level agreement (SLA) services.
- **MEC Services**: Their goal is serving all MEC applications and use cases, providing the foundations for implementing network applications in a distributed and abstracted scope. Positioning, KPI monitoring and provisioning, IP and named data services, event logging, general analytics/logistics and network configuration are vital examples.

Services that are deployed by their respected physical MEC Hosts, are essentially supported by application-specific software instances that are ***Virtual Machine-based*** (VM-based) e.g. KVM or ***Container-based***. As of the latter case, various containerization techniques can be found, such as *system containers* (e.g. LXC, LXD), where the user-level environment mimics a fully featured OS, or *application containers (e.g. Docker),* where the container is typically expected to host a single user process [23].

VMs play a crucial role in Datacenter Resource Management. A VM serves as a logical container, hosting software instances with data and can run using only a subset of a server's Hardware resources. This means that full virtualization via hypervisor takes place, fully emulating the OS kernel and hardware. In the container case, kernel and hardware are shared. This means that containers require less computing resources and lower virtualization overhead [16], but their host-sharing nature means that containers are generally less adaptable (e.g. A Linux Container cannot run on a Windows server). A possible remedy for this, is nesting the containers inside VMs [13].

Another note is on the security issues around containers, since VMs offer complete isolation from their physical hosts and from other deployed VMs on the same machine, being immune to an attack from a compromised component. Security is an important prerequisite for any enterprise software and with Containers, there is a serious concern on this. Thus, we result in using VMs for deployed MEC Services in our setup, despite the important downside of having to transfer high volumes of data over the network in the case of Service Migration, which will be extensively covered in later sections. Noted, though, is the ability to dissect the Virtual Host's OS into Base (vanilla OS distribution), Application (modules, files and programs) and Instance (Application State) Layers in order to aggressively reduce the data load to be transferred [13].

3.3: MEC Platform Placement

As of the areas of placement and workload management, additional issues require to be addressed. Commonly, MEC Units reside at the LTE eNB or at a multi-RAT cell aggregation site. The latter can be located indoors within an enterprise e.g. a hospital, or indoors/outdoors, depending on the deployment use case, usually for stadiums, malls or generally public/private areas, characterized by a large number of connected users.

MEC units can be therefore collocated with any network node, starting from the Backhaul/Core, down to the Midhaul/Fronthaul. The nearer to the Core, the more reachable to Network Users, but the higher the Latency in User-Service communication. *Service Latency* is essentially the Round Trip Time (RTT) from sending a request to getting a response, divided into Transmission, Propagation, Queuing and Processing times.

Generally, MEC platform placement depends on various factors, such as scalability, physical deployment constraints, performance requirements and the network information to be exposed. That means that in some scenarios, MEC Services may not be available or applicable.

A commonly suggested practice in terms of MEC platform placement, to bring its deployed Services closer to the Network, is the *bump in the wire* approach. Data exchanged between the Base Station and the Core Network (S1AP traffic – backhaul link) is intercepted and redirected to the MEC Service. Furthermore, under the Cloud-RAN paradigm, ETSI has reformed this approach into collocating the MEC ecosystem with the CU component at an Edge datacenter, again, practicing S1AP traffic interception. For a user to access a deployed MEC Service, the resulted traffic path is UE-DU-CU-MEC.

This path can be further shortened to UE-DU-MEC, efficiently utilizing our aforementioned PDCP/RLC base station functional split, in order to place MEC on the fronthaul of the CU node (or equivalently the midhaul of our disaggregated Base Station). Since this split option uses Ethernet-based encapsulation, CU-fronthaul-based Services can easily manage the related traffic. Other placement options include using a distributed Core Network to control the MEC Services via the P-GW components.

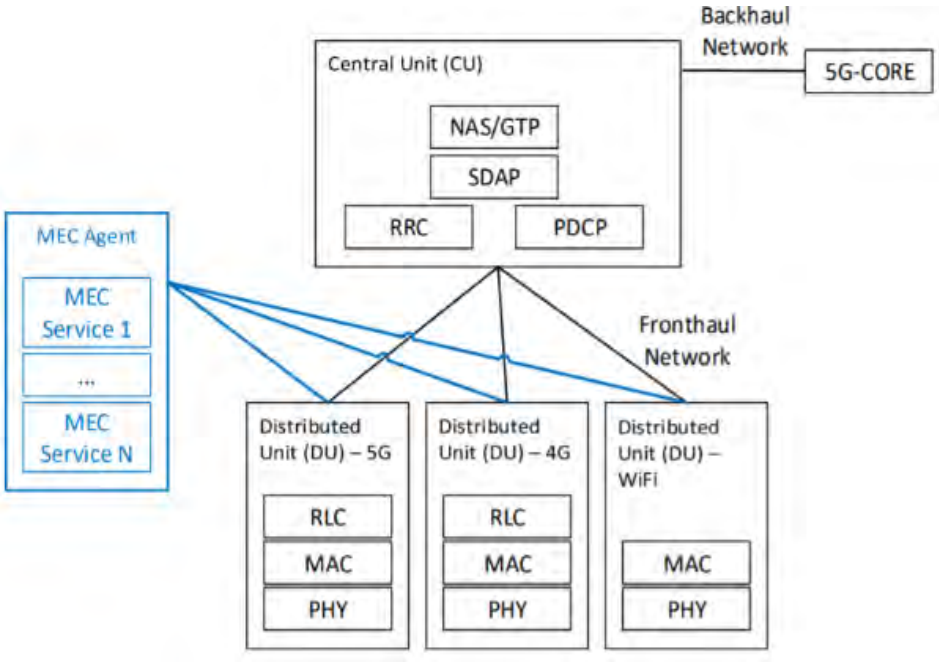

Figure 12: MEC Platform placement on the fronthaul of Heterogeneous 5G Cloud-RANs

To elaborate, the above system is comprised of elements that orchestrate the CU – DU communication, a MEC Agent and an internal mapping system.

MEC Agent operates in a similar philosophy to the F1oIP protocol agent, which instrumented the CU-DUs communication in section 3.3, handling the delivery of the User-Service traffic and communicating with the multi-technology DUs. It has the ability to generate and exchange appropriate messages with the DUs or receive and deliver to the MEC Host-deployed Services. Following the Data Request/Data Indication axiom that was presented previously, a *mec_data_request* message is generated for the DU-MEC path and a *mec_data_indication* is generated for sending to the dynamically-discovered (from the initial DU-MEC exchange) DU that our Service is registered to. The first message type is picked up by the Agent, which handles the encapsulated user data packets and delivers them to the corresponding Service. In the latter case, the agent generates this message to send to the dynamically-discovered DU that the current client is registered with.

The mapping system's functionality includes the linking of UEs to their respective low-level Layer 2 credentials. Those credentials are the Radio Network Temporary Identifier (RNTI) in the case of cellular and MAC addresses for the WiFi case.

Cellular base stations are essentially seen as L2 devices from the UE perspective. That means that a UE establishes a connection with the Core Network in PDN context. Each PDN enables a separate broadcast channel and all downward network components can communicate with each other i.e. the Core Network and the UEs. As the MEC Agent only interfaces the Distributed part of our RAN, data from cellular technology is only interfaced through low-level L2 information, in the form of RNTIs, used by the base stations for forwarding the User Plane data and mapping to different Logical and Transport Channels. As prior mentioned, the cases of WiFi and MEC Services can be solely interfaced by their respective IP address. This lead to the design of a new signaling procedure that broadcasts a

new client's credentials from all available interfaces (IPs for WiFi & RNTI for cellular). Upon a client's registration with the network, this context signaling operation takes place, making users' presence in the network transparent to all involved components.

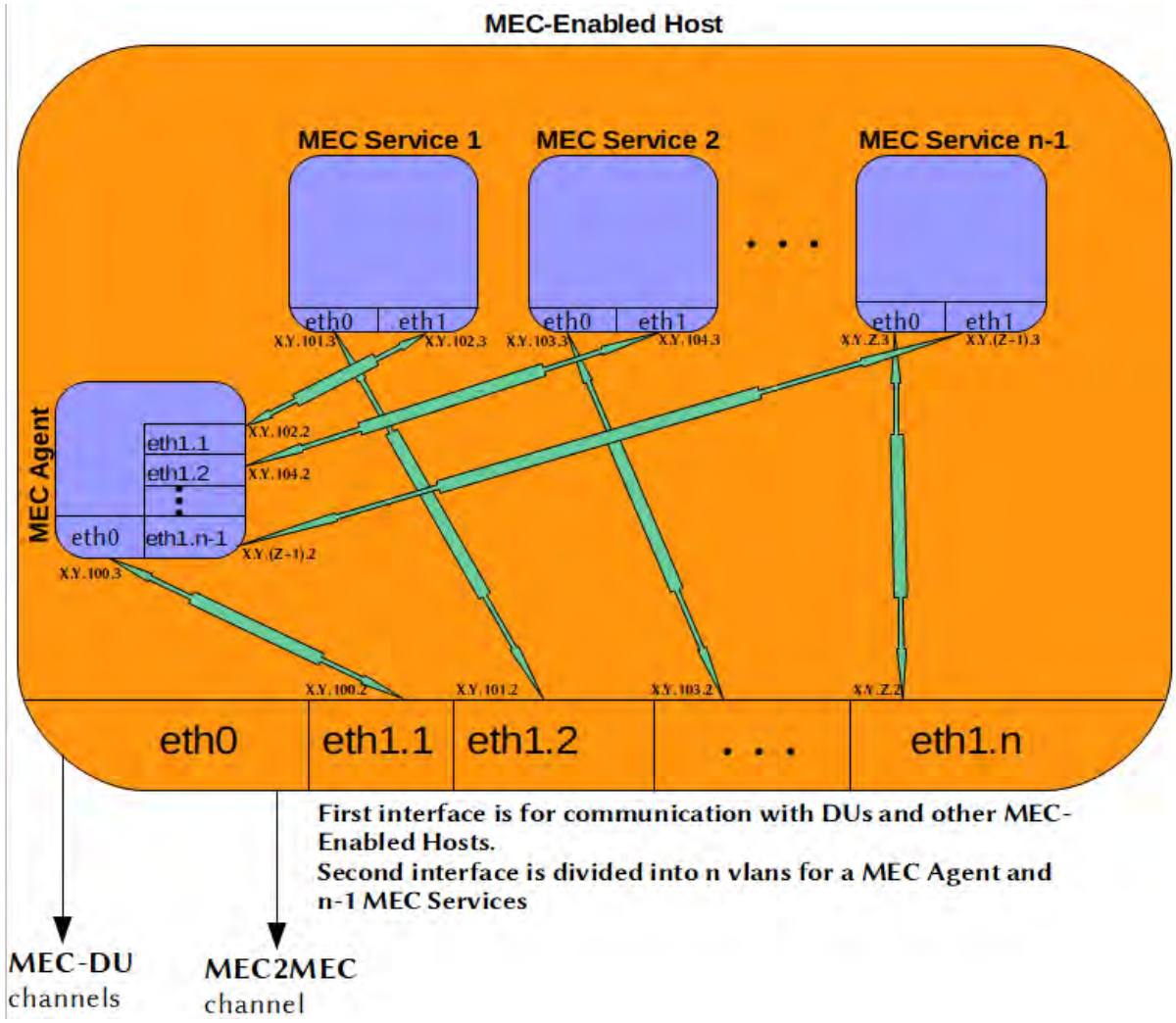

Figure 13: MEC-Enabled Host schematic

In a nutshell, a MEC-Enabled Host is depicted, along with its deployed KVM MEC Agent & MEC Services. For each service VM, an additional virtual bridge is created, connecting the Host's appropriate virtual interface to the virtual Host's corresponding interface. Additional virtual bridges are also present to actualize the communication between the agent and the services.

Chapter 4 – V2X and MEC

4.1: MEC-Enabled V2X

As connected car numbers increase, so is the volume of data aired and the need to minimize latency. Centric structures for homing the data that is generated may or may not be a suitable design choice, depending on the use case tackled. Vehicular network communications require low latency. Current LTE deployments can be adequate for real-time distribution of use case–specific messages, and can also complement Digital Short-Range Communications network deployments. MEC's role in this context is to extend the connected car cloud into the distributed Base Station environment and create a layer of abstraction from both the Core Network and the applications that are provided over the Internet.

By stimulating the collaboration of devices on the edge to offload computational and communicational tasks, the link connectivity and Quality of Service indications in MEC-Enabled Vehicular Networks can be quite leveraged. Vehicles, serving as edge nodes in this concept, contribute to real-time management of traffic, by helping in minimizing the average response time of the reported events by connected vehicles. Events that occur, such as traffic jams, road hazards/accidents can be recorded by one edge node in the format of image, video or generally a context-informative message, that can be uploaded using one of the RATs available to the user, depending on the policy imposed for technology selection. V2X communications will be further enriched with value-added services, such as car parking space tracking and Infotainment (e.g. Video Distribution). Based on the GS MEC 012–specified Radio Network Information Service (RNIS) which shall provide actual RAN information related to UEs [42], future extensions aim at providing V2X applications with predictions on the QoS performance of the V2I (Vehicle to Infrastructure) link [19]. This will be the focus of later sections and implemented functionalities.

4.2: 5G MEC-enabled V2X

MEC-enabled Hosts can deploy multiple services that belong to the same or to multiple categories of the MEC Service ecosystem. A user characterized by mobility is connected to one or multiple MEC Services, enjoying low-latency and high-bandwidth access to multimedia and/or context-awareness information (e.g. Video Streaming/Road Safety Applications), with an ability to be served via multiple RATs.

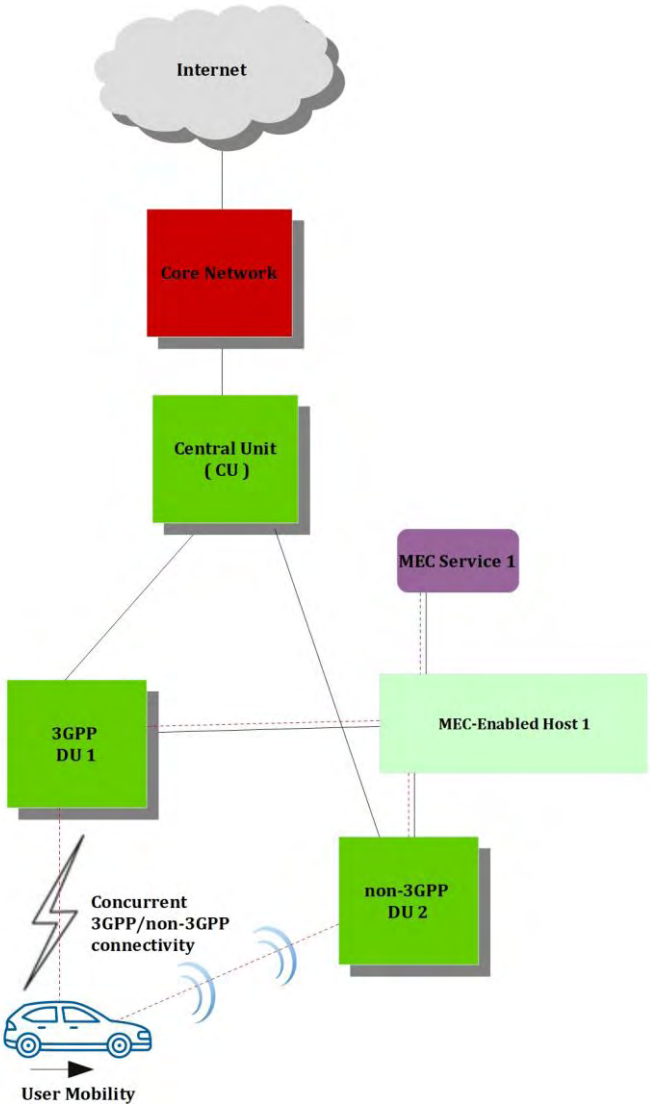

Figure 14.1: 5G MEC-Enabled V2X Network

4.2.1: Radio Access Technology switch

In our dual-technology setup of 3GPP and non-3GPP technologies, as the release of a stable NR version is still imminent, a UE can maintain connectivity through both. All content that is aired either in the DL or UL case, is managed by the essence of a Radio Access Technology (RAT) switch. This controller has the ability to interchangeably utilize the links between the MEC Host and the multi-Technology DUs to schedule the Service<->User traffic i.e. the traffic can follow either of the MEC<->3GPP DU<->UE or MEC<->non-3GPP DU<->UE paths. On command, the technology configuration for a specific user can be statically altered on any time instance during *uptime* (the time in which a service is up and running).

4.2.2: A practical example of the need for a Migration Scheme

An example of one of the use cases that need to be tackled by our setup's functionalities can be presented, beginning with Figure 14.2. The mobile multi-homed UE continues his/her trajectory, gradually moving away from the 3GPP DU's coverage. The degradation in the initially 3GPP-served, user-perceived QoS, can post-event or near-time trigger a RAT switch, so as the user's DL traffic is subsequently served through the MEC<->non-3GPP DU<->UE path, thus offering the best latency available.

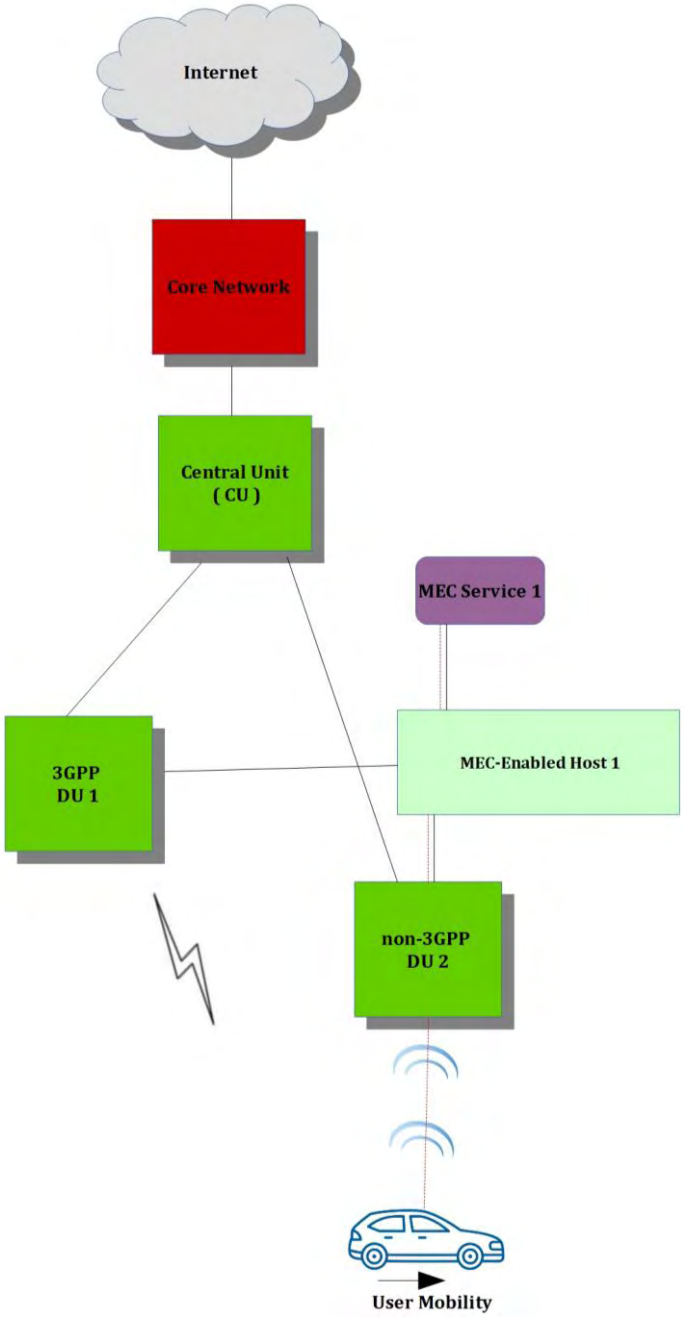

Figure 14.2: Computational offload via WiFi DU, after a RAT switch

Later in time (Figure 14.3), the QoS of the user served through non-3GPP technology e.g. WiFi is also degraded (either due to congestion-related issues, or simply by moving away from both DUs). The imminent disruption of service and the gradual increase in the perceived service access latency, creates the imperative need of service migration schemes.

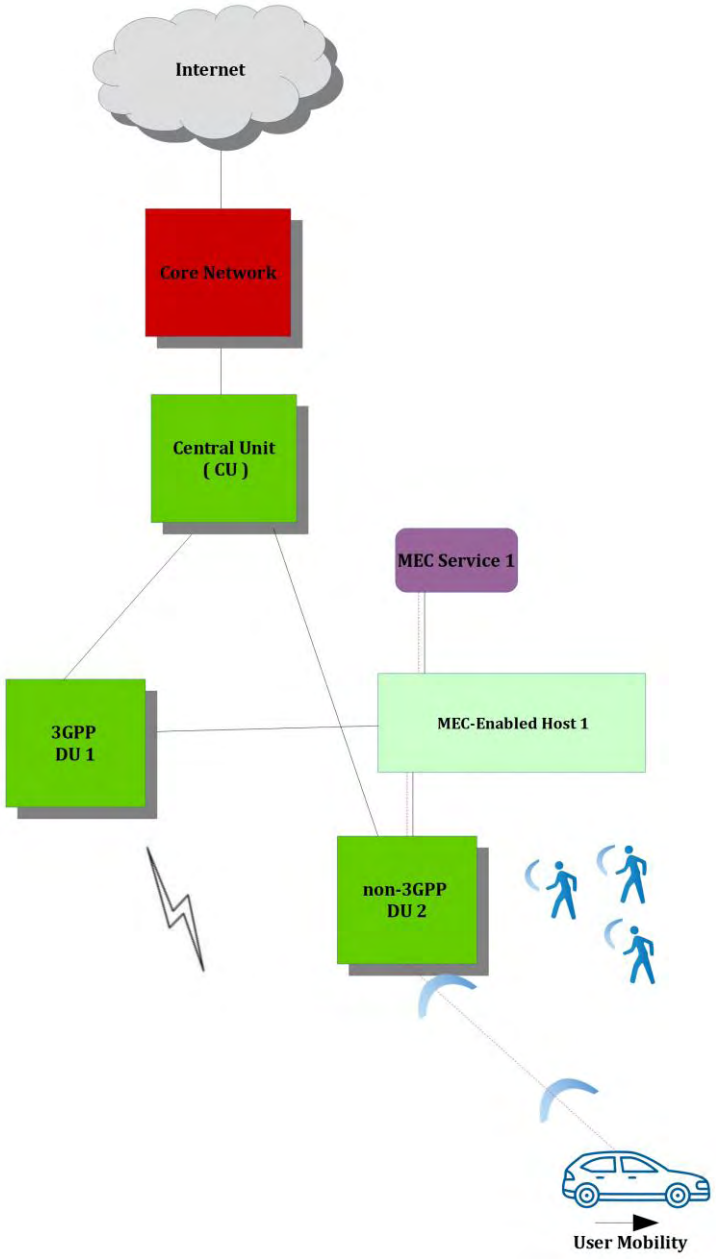

Figure 14.3: Degradation of QoS in both Technologies

Typically, networks perform inter-eNB Handover schemes to switch to a neighbouring RAN (Figure 14.4), instead of the one that the user is currently offered connectivity from. That means that the MEC Services need to also mitigate to a new location in a non-disruptive manner.

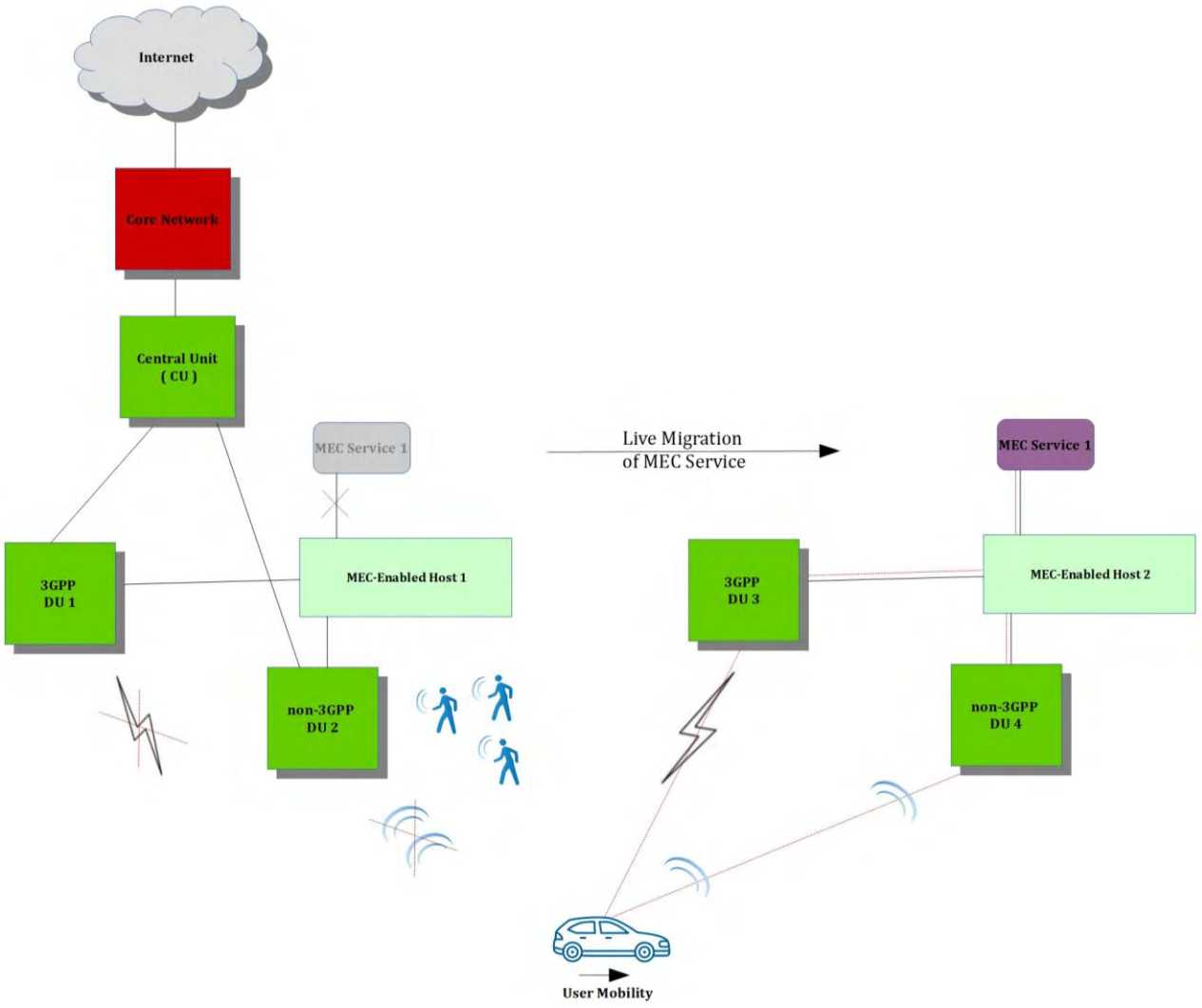

Figure 14.4: Inter-eNB Handover and Live MEC Service Migration

The user above reaches the vicinity of a secondary deployed RAN (also a secondary MEC-Enabled Host) and is able to initiate the inter-eNB Handover and MEC Service Migration schemes. When a user changes its attachment point from one MEC Host to another, migrating active services is required for continuous, seamless service support. It is beneficial for the MEC Services to be migrated from the initial host to the new one, actualizing a *follow-me approach*.

4.2.3: On MEC Service Live Migration

A user is offered, continuously and uninterruptedly, access to a MEC Service for a period of time. At some point a need for the Service to be migrated occurs.

Live migration of a Virtual Machine improves utilization of resources, load balancing of high-processing power nodes, fault tolerance in virtual machines, all with the aim of increasing portability and physical server efficiency. Migrating a service in a live manner, involves transferring the VM's memory state data to the destination machine [22]. This procedure focuses on:

- Minimizing the *downtime* aspect i.e. how much time a user *cannot* receive service, contrary to edge datacenter service migration, which aims in reducing total time of completion of the migration scheme, as end-to-end latency can be impaired in the process. Recalling the 0ms downtime requirement of chapter 1, we can realize the importance of implementing an always-available and nomadic MEC ecosystem.
- Independence on the availability of a dedicated computational unit or high-bandwidth network i.e. overcoming high variation of network bandwidth and computation capacity caused by workloads that vary across time.

An ideal copy of a virtual machine is its complete state, including *memory, disk* and any *established network connections*. Local disk and network interface migration are not trivial in being carried out successfully.

In order to maintain connectivity after the completion of this process, it is vital to preserve open network connections, so as network clients can be attended with as close to no service disruption. If migration occurs within the same LAN (Local Area Network), a VM should retain its original IP after completion, by generating an *unsolicited ARP reply* (broadcasting its link layer credentials) so as its new location can become reachable. In a WAN (Wide Area Network) migration, utilization of techniques as VPN (Virtual Private Network), *Tunneling* and DNS servers can be quite effective. Some cases outside WAN context exclude disk migration, assuming a SAN or NAS strategy. Briefly, *Storage Area Networks* typically use fiber channels to interconnect and connect a set of storage devices that share data. *Network Attached Storage* units include a dedicated hardware device, connected to LAN. This NAS server authenticates clients and manages file operations as a commonplace NFS which runs an embedded OS.

Migrating the disk component usually takes up most of the total migration time, ranging up to hundreds of Gigabytes [24].

4.2.3.1: Stateless vs Stateful Migration

Stateless migration does not transfer application running states, only redirects the requests by the user to the destination server, which is running a separate instance of the Service. This case is only applicable for services that do not keep track of users' states.

Stateful migration is initiated and carried out seamlessly, while the user continues receiving Service. At completion, the target machine has gathered all necessary user state to continue from where the previous machine left off, without breaking down any previously established connections.

Due to latest demand for interactive services, which are our main focus, such as mobile video streaming, Road Safety applications and mobile online gaming, keeping state for the user is essential. Thus, we result into a stateful migration scheme.

4.2.3.2: Pre-copy vs Post-copy Migration

Two categories of handling the VM memory state data transfer from the source to the destination hosts are pre-copy and post-copy schemes.

In **Pre-copy Migration**, all memory pages from the source host are duplicated while the VM is in a running state. Pages modified during this procedure, called *dirty pages*, are copied again, until the ratio of re-copied pages is higher than the one of changed pages. Next, the instance on the source is *stopped,* the remaining pages are transferred to the destination and the VM instance is resumed in the destination. This method can reduce downtime with the tradeoff of having to transfer more data than post-copy migration.

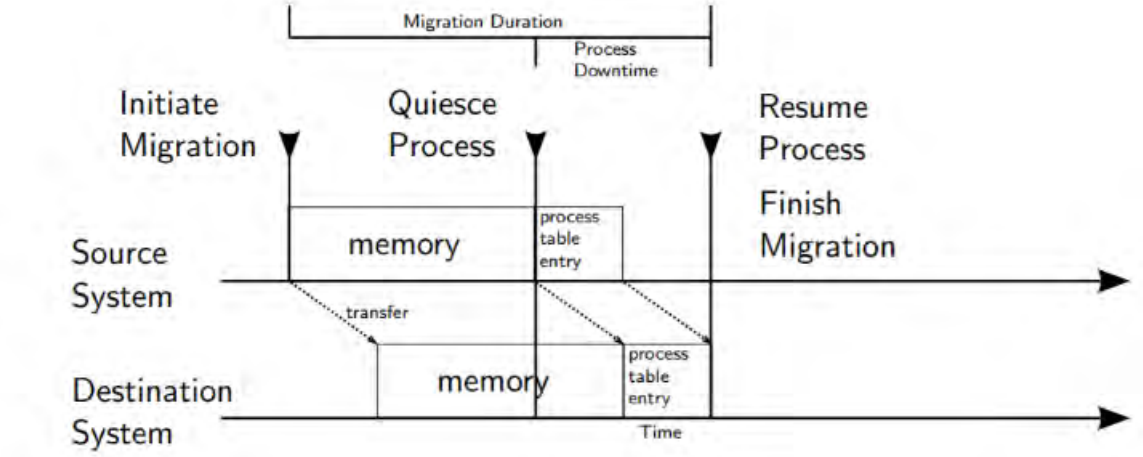

Figure 15.1: Pre-copy Migration

Post-copy Migration is initiated by suspending the VM instance that is running on the source physical host. A minimal subset of memory state is then transferred to the destination (e.g. CPU state, registers, non-pageable memory), followed by the restarting of the instance in the destination. This method transfers less data but may result in longer downtime.

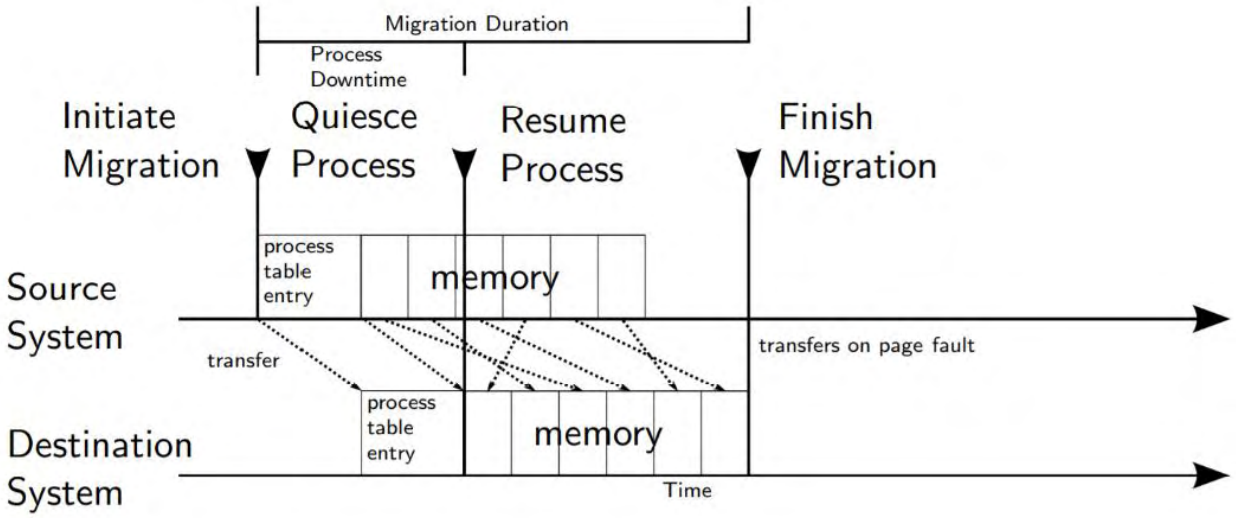

Figure 15.2: Post-copy Migration

4.2.4: Migrating with KVM

Kernel-based Virtual Machine (KVM) is a virtual machine monitor which enables full virtualization of x86 hardware, essentially making possible the deployment of multiple unmodified Linux or Windows operating systems.

Live KVM migration utilizes the pre-copy approach and performs record-keeping of dirty pages for scheduling their transfer to destination. The latter is used as a bitmap of modified pages since the last call. KVM initially maps the virtual Host's pages as read-only and after the first write access, it maps them for write. The algorithm is divided in 3 phases in an iterative fashion. In its first phase, all memory pages are marked as dirty and the modification tracking mechanism is initialized. Next, pages marked as dirty are copied to destination iteratively, until the maximum transfer rate to the destination machine is not exceeded. Dirty pages that are not yet copied, are used as a means of estimating the consequent downtime on the third phase initiation. If this estimation yields a high value compared to the target value, the procedure iterates until it predicts a lower and suitable one. On reaching the target value, the third and final phase is started, where the source machine and applications are suspended until all remaining dirty pages are transferred to destination, all registers are loaded and execution on the new host is resumed [24].

KVM has its own User Interface, *virsh*, for managing its deployed Virtual Machines. It offers a wide variety of options and configurations on the migration procedure, including peer to peer migration, migration via IP tunnels, encryption and compression, incremental and whole storage copying schemes, etc. Experimentation around those settings on a fully-fledged 5G MEC-Enabled Heterogeneous setup, as outlined, showed that the live migration completion time of a MEC-deployed VM with virsh, ranges from 4 to 7 minutes, concerning interactive services placement. While this may seem restrictive, we should remind that our

focus is true zero downtime and no service disruption/critical performance degradation. Indeed, the user terminal perceives virtually no disruption, maintaining any open connections without breaking down, while performance on a 1 Gbps link degrades about 10% for a fraction of overall migration time that corresponds to a few seconds. Disk migration is included in the process and is the main factor of the above migration times.

The load to be transferred can be cut down aggressively via OS dissection, as prior noted, leading to effectively reduced total migration times, while combining it with a SAN/NAS scheme can lead to having to transfer only the Instance Layer of section 3.2. Factors affecting migration speed, thus migration time include network speed, workload, number of folders to be transferred, network latency.

Not typically encountered, but prohibiting case of live migration is when the VM of interest continuously modifies pages in a rate that is faster than KVM can transfer. In this case, offline migration is recommended. Choosing the network interface with the highest throughput for migration data is recommended, thus a fiber-based MEC2MEC channel may ensure a timely transfer at a relative cost.

Migration options used in our migration scheme with virsh are data compression, P2P migration and *tunneling over SSH* connections on the MEC2MEC link.

4.3: Closing Remark on 5G MEC-Enabled V2X

MEC's aforementioned benefits of 1-hop Cloud access, with pioneering latencies and data rates, along with reduced generated traffic per UE transmission have led to ETSI-driven standardization initiatives. During the conception of this thesis, MEC solutions for 5G V2X applications are at Phase 2 and seem promising to be evolved to completion. Its adoption in a future Heterogeneous Cloud-RAN will render as baseline, instant and ubiquitous user access to the Network, while being offered a personalized experience that is characterized by individually-built Intelligence, spatio-temporal context awareness and proactive decision-making. Moreover, it will provide guaranteed zero downtime with best-RAT-offered latencies in a real-time or, most importantly, in a proactive manner, thus fulfilling a truly next-generation role and purpose in our ever-evolving Access Network ecosystem.

Chapter 5 – Vehicular User QoS across time and space

To build around the concepts of context awareness and prediction for a future 5G MEC-Enabled Vehicular Network, we first need to study, observe, and decide, upon obtaining data point measurements. As most of modern-era technological deployments dictate, data streams are continuous in our ever-expanding samples. This gives a great chance to the field of Mathematics to provide us with powerful tools in order to infer on and model via Machine Learning and/or Neural Networks the collected data. Indeed, Hypothesis Testing, uncovering Dynamics between different factors, linear and non-linear modelling, mathematical transformations and the incorporation of time and location information, creates a plethora of possible approaches for, literally, landscaping the future.

Borrowing from concepts and practices found in Mathematics and Econometrics literature, we implement, visualize and test on streaming *real-world data*, which need careful handling and utilization, while imposing practical and timing constraints.

This chapter will first introduce our methodology for proactive decision-making. Next are metrics of interest and their behaviour across time. Then, methods to infer on and modify their statistical properties are, as carefully as possible, introduced. An exploratory of Data Analysis, Modelling and Predicting both spatially and temporally is presented and a section-specific walkthrough leads us to our final framework, which combines the freedom in easy development and the packaging of complete mathematical methods provided by Python and R, respectively.

5.1: Proposed Proactive Decision Schemes

As outlined, we choose to approach the User QoS provisioning matter through different methodologies, employing time windows for the temporal study and interpolating over a spatial grid for the spatial case. We suppose that an underlying *Data Generating Process* (DGP) can be modelled and utilized on measures of interest, either for *forecasting h time steps* ahead in time or interpolating over a spatial grid of fixed resolution.

A time window scheme typically deals with searching for the optimal sequence of actions that minimize the average cost defined over a given time window. A *look-ahead window* is defined as a time period in the future that can be predicted, while *uncertainty measures* alongside the predictions e.g. confidence intervals or standard deviations are strongly recommended. In our context, minimizing the cost means limiting losses in combined user bandwidth, as induced by non-optimal technology switching and MEC Services migration. Measures of interest are practically the reported user QoS measurements, used to replicate the user's view of the network to the network itself and being a direct indicator of achievable rates and bandwidth. Time window-based decisions can efficiently tackle a more general setting, in terms of network structure and mobility patterns. Such an example is found in a recent white paper on 6G networks [28]. Look-ahead window size can actually be perceived as a factor of our decisions consistency, since using too large a size can result in higher prediction errors and consequently, to more frequently wrong decisions made, concerning placement of MEC Services [22] and RAT switching.

Under complete coverage of a user in the RAN vicinity, owing to the envisioned deployment density of DUs/RSUs (Road Side Units) and efficient cell reselection schemes, we can expand our data across each domain, while spatio-temporal models are also noted as an interesting approach. The information used as input to the logic implemented, is a problem hyperparameter: the sample size. A sample consists of on-instance fetched datapoint measurements, used as input to an algorithm that expands the sample's asymptotically-calculated statistical properties, across time and space. Samples can be filtered appropriately, to create informative and flexible windows.

The window flexibility factor can be controlled by the granularity of the per-user measurements, in conjunction with the freshness of the information that we choose to impose. Very frequent per-user measurements and frequent discarding of data seem appropriate. An information aging scheme e.g. AoI (Age of Information) [30], can be employed for intelligent aging policies.

Increased sample size can prove beneficial for spatial analysis and interpolation, since more points have measurements in a given grid, so higher accuracies can be achieved at the tradeoff of higher computational complexity. For the case of time forecasting, more points can or cannot be more useful for better predictions, depending on the behaviour of the captured QoS windows across time. While employed models may result in variable prediction accuracies, they can optimistically provide an upper bound to the losses occurred by inelastic present schemes i.e. the cost of a wrong prediction is low, while a good prediction has an immediate impact, instantly benefiting the user from the QoE standpoint. Furthermore, we can affect the Plasticity vs Stability tradeoff in our predictions [29] by exposing the posed hyperparameters to per-user or/and global prediction error monitoring processes that can real-time modify the utilized sample size, look-ahead window size and spatial interpolation resolution in an online manner.

RAT Switching is to be studied on the time dimension in a personalized manner. A window of fresh and preprocessed per-user QoS measurements is fed to user-dedicated processes that extract apparent dynamics, decide on the window's statistical properties, perform model selection and estimation, followed by multi-step time forecasting. This simulates the user's expected QoS while having active connections to 3GPP and non-3GPP technologies on accessing multiple MEC Services. In such way, we allow for intelligent scheduling of the Data Plane across the look-ahead window and minimize the losses in combined user bandwidth.

Migrating MEC Services can be studied both on the temporal and spatial dimensions. On the former, it means predicting persistent non-tolerable QoS for both type of technologies, suggesting the initiation of the migration procedure. We will focus on the latter i.e. utilizing QoS measurements from *crowdsourcing* schemes of multiple similar users (e.g. in terms of equipment capabilities/specifications, relative location/trajectory, etc.) in a RAN, to create frequently updated and available-online Radio Environment Maps (REMs) [31], attempting to provide a wider view of the QoS in a RAN, aiding placement.

5.2: QoS Metrics Considered

Typically in cellular networks, mobile users execute a variety of *handovers,* essentially migrations of their radio context and state, in cases of serving cell selection/reselection procedures that are usually triggered by QoS metrics thresholding. Such measurements, which support important features as inter-RAT mobility, are reported to the serving eNB in RRC_CONNECTED mode. In RRC_IDLE mode, the measurements are not reported but may be used autonomously by a UE for cell reselection. Usually based on Reference Signals, while some are generalizations over a wider spectrum, they can nonetheless be collected in a systematic manner for examination, to be used by our case.

RSRP (Reference Signal Received Power) is measured by a UE over the cell-specific Reference Signals (RSs) within the measurement bandwidth, over a measurement period. It is a type of signal strength measurement and is indicative of the cell coverage. It is defined as the linear average over the power contributions (in Watts) of the Resource Elements (REs) that carry cell-specific Reference Signals within the considered measurement frequency bandwidth (*REs* are the equivalent of 1 modulation symbol on a 15 kHz subcarrier i.e. 2 bits for QPSK, 4 bits for 16 QAM and 6 bits for 64 QAM). Applicable in both IDLE and CONNECTED modes, it is utilized for cell reselection within the same RAN (intra/inter frequency handovers) and is fundamental in being included in our study. Its theoretical range is [-140,-44] dBm, while it can be reported in integer form with 1 dB resolution, ranging from 0 to 97.

RSSI (Received Signal Strength Indicator) measures the average total received power observed only in REs containing Reference Signals in the measurement bandwidth over N Resource Blocks (RBs), which translates in the total received *wideband* power (RBs are 12 consecutive 15 kHz subcarriers in the frequency domain (180 kHz) x 1 slot period in the time domain). The total received power of the carrier RSSI includes the power from co-channel serving & non-serving cells, adjacent channel interference, thermal noise, etc. Its theoretical range is [-120, 0] dBm.

RSRQ (Reference Signal Received Quality) is the ratio of RSRP to RSSI for a specific carrier and is measured only in connected state. The RSSI part of this, is the total received power including interference from all sources, as mentioned above. It is calculated in a similar way to RSSI, using REs that contain RSs for antenna port 0 within the measurement bandwidth. The interference component of RSRQ quantifies the received signal quality considering both signal strength and interference, which may vary, according to the measuring UE's location. Combined with RSRP, they find their use in intra-frequency quality-based handover schemes. This metric is a relative quantity and can ease, in some extent, from absolute measurement errors. Its theoretical range is [-19.5, -3.0] dB.

RSSNR (Received Signal to Noise Ratio) is the ratio of the RSRP metric to the noise power of a 15 kHz subcarrier in dB, under the assumption that RSRP does not contain noise power and for a predetermined noise figure and temperature values. Its theoretical range is unspecified since it depends on the underlying network deployment and technologies employed, while practically ranging from sub -100 to over +200 dB. RSSNR is lowest at the edges of a cell, owing to inter-cell interference from the neighbouring BSs [26]. Typically, scoring higher than 20 dB, means that the UE is offered a relatively good QoS from the metric's standpoint.

Transmit power varies in levels, owing to factors such as proximity to BSs, the number of antennas that a device employs, signal noise/interference. Concerning cell selection procedures, if they are based solely on RSRP readings, they can create a problem of load imbalance. If only e.g. RSRP or SINR (Signal-to-Interference Noise Ration) is considered, users prefer connecting to network components of higher transmit power e.g. macroBSs, leading to their overloading/overutilization, while under-loading small cells. Works like [19], showcase that considering multiple factors (such as allocated bandwidth, processing power, distance between endpoints, geolocation and noise power) can form efficient cell reselection procedures that is an important prerequisite in 5G Heterogeneous Networks. A multivariate approach is also employed in our study, because the history of one metric alone can lead to missing data points e.g. on the case of falling back to a 3GPP legacy technology in LTE (during a HSDPA fallback for a voice call, there is no RSRP metric available). We continue in handling the mentioned 3GPP-specific metrics as a set of moderately-correlated factors [54], due to the underlying theory that was showcased. For the non-3GPP case, we also incorporate *WiFi-specific RSSI* in our study, to quantify the relative QoS through non-3GPP channels.

5.2.1: Drive Tests Data Collection

In order to study the aforementioned metrics' behaviour across time and to visualize the QoS evolution during drive tests of a single/multiple vehicle users in a dual technology setting, a lightweight and efficient recording tool has to be employed. Various network session parameters can be extracted from an Android device via a Java-based QoS parsing and monitoring application. This application utilizes the *oml2* measurement framework, to efficiently package and record-keep these informative instances, while incorporating timing, geolocation and other information such as technology used, user to service latency, etc. In this context, a mobile UE in the network runs this application continuously in client mode and connects to the oml2 server side, which adds timestamps and sequence numbers, as well as sender identification. The server side can be typically deployed on a machine that resides anywhere in the network, from a RAN component to the MEC-Enabled Host, homing a remotely accessible PostgreSQL Database, in which the formed *measurement points* are injected. The resolution of the measurements in time is fixed, as below on a less-restricting case of 6 secs. A first faced technical restriction is the incorporation of both technologies' QoS measurements for vehicular users, alongside geolocation information that forms a trajectory. As our testbed experimental setup does not permit for UE mobility, to approach each dually-connected user characterized by mobility, we used two identical mobile devices that were placed in the same vehicle during the drive tests, where the one provided non-3GPP connectivity to the other. Multiple drive tests for the Urban Area case were conducted in the city of Volos, Greece, concerning real trajectories of moderate-speed users.

	oml_ts_server	mRssi	mRsrq	mRsrp	mRssnr	loclat	loclon
0	2020-07-19 19:46:16	-53.0	-12.0	-83.0	72.0	39.37303	22.90867
1	2020-07-19 19:46:22	-51.0	-12.0	-85.0	94.0	39.37272	22.90965
2	2020-07-19 19:46:28	-51.0	-12.0	-84.0	84.0	39.37244	22.91040
3	2020-07-19 19:46:35	-57.0	-13.0	-87.0	-10.0	39.37219	22.91103
4	2020-07-19 19:46:41	-51.0	-13.0	-91.0	44.0	39.37200	22.91147
5	2020-07-19 19:46:47	-57.0	-12.0	-91.0	58.0	39.37197	22.91155
6	2020-07-19 19:46:53	-57.0	-12.0	-91.0	24.0	39.37198	22.91154
7	2020-07-19 19:46:59	-59.0	-12.0	-93.0	60.0	39.37197	22.91154
8	2020-07-19 19:47:06	-55.0	-12.0	-92.0	86.0	39.37194	22.91161
9	2020-07-19 19:47:12	-59.0	-13.0	-93.0	70.0	39.37177	22.91203
10	2020-07-19 19:47:18	-61.0	-12.0	-93.0	28.0	39.37146	22.91279
11	2020-07-19 19:47:24	-63.0	-12.0	-93.0	30.0	39.37116	22.91354
12	2020-07-19 19:47:31	-59.0	-12.0	-93.0	36.0	39.37081	22.91438
13	2020-07-19 19:47:37	-61.0	-13.0	-93.0	2.0	39.37045	22.91533
14	2020-07-19 19:47:44	-65.0	-13.0	-95.0	10.0	39.37017	22.91605
15	2020-07-19 19:47:50	-63.0	-12.0	-96.0	-20.0	39.36988	22.91677
16	2020-07-19 19:47:56	-65.0	-12.0	-96.0	-2.0	39.36954	22.91762
17	2020-07-19 19:48:02	-61.0	-13.0	-96.0	-38.0	39.36919	22.91839
18	2020-07-19 19:48:09	-57.0	-15.0	-97.0	48.0	39.36882	22.91912
19	2020-07-19 19:48:15	-63.0	-14.0	-96.0	50.0	39.36843	22.91984
20	2020-07-19 19:48:21	-63.0	-13.0	-96.0	18.0	39.36803	22.92061
21	2020-07-19 19:48:27	-59.0	-13.0	-95.0	14.0	39.36768	22.92122
22	2020-07-19 19:48:34	-59.0	-12.0	-91.0	96.0	39.36736	22.92184
23	2020-07-19 19:48:40	-55.0	-8.0	-84.0	110.0	39.36703	22.92247
24	2020-07-19 19:48:46	-53.0	-7.0	-79.0	162.0	39.36686	22.92274

Figure 16: A minimal example of the collected measurement points for a single User

5.2.2: Basic Preprocessing

Our set of measurements in the time domain essentially forms a Multiple Time Series (MTS) problem. That means that the natural ordering of the measurements must be preserved in order for them to be efficiently studied. Upon receiving the raw-recorded data, some simple preprocessing steps have to be carried out, in order to proceed in a smooth and non-error-prone manner. That is:

1. Imposing the correct data types and representations: This step involves converting from Python-style timestamps to UTC-formatted time, typecasting values to their correct format e.g. 64-bit floating point precision on the QoS metrics, optional projection of the spatial coordinates that come in longitude/latitude form, etc.
2. Erroneous/missing measurements detection & imputation: Typically, erroneous metrics are values that either fall outside of their theoretical range or have invalid values to account for their absence e.g. 0 for a coordinate or -1 for latency. These can owe to various factors like sensor/software malfunctions. They can be either completely discarded e.g. in the case where we have a sufficient sample size or ordering is irrelevant, etc. These errors can typically denote absence of measurement until a procedure completion, while timestamps and geolocation are important for us to discard. Thus, erroneous values are first set to NaN, then interpolated via OLS (Ordinary Least Squares) regression, including rows and/or columns as predictors via *MICE* (Multivariate Imputation via Chained Equations) method.

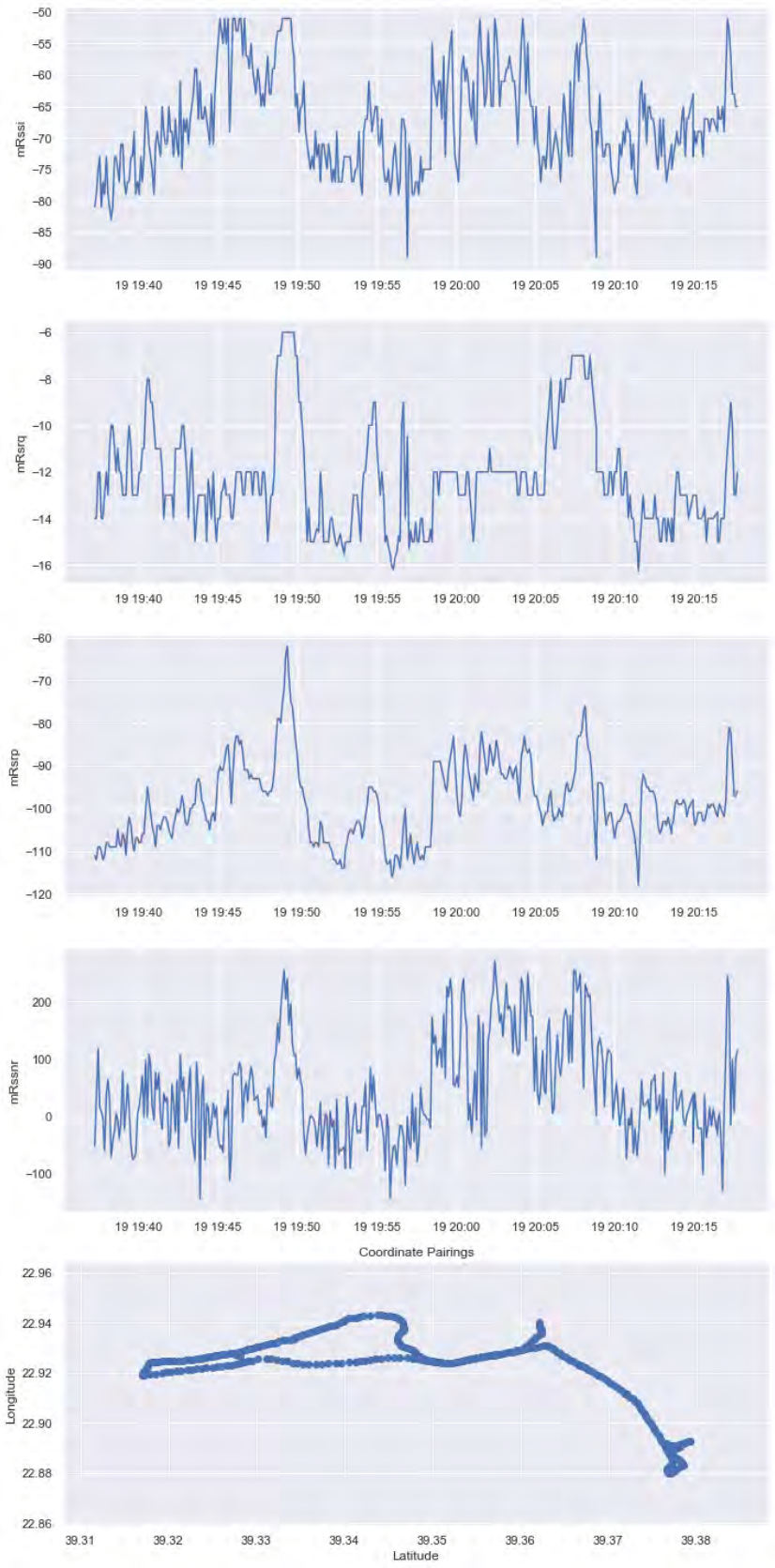

Figure 17: Example behaviour of a single user QoS measurement excerpt

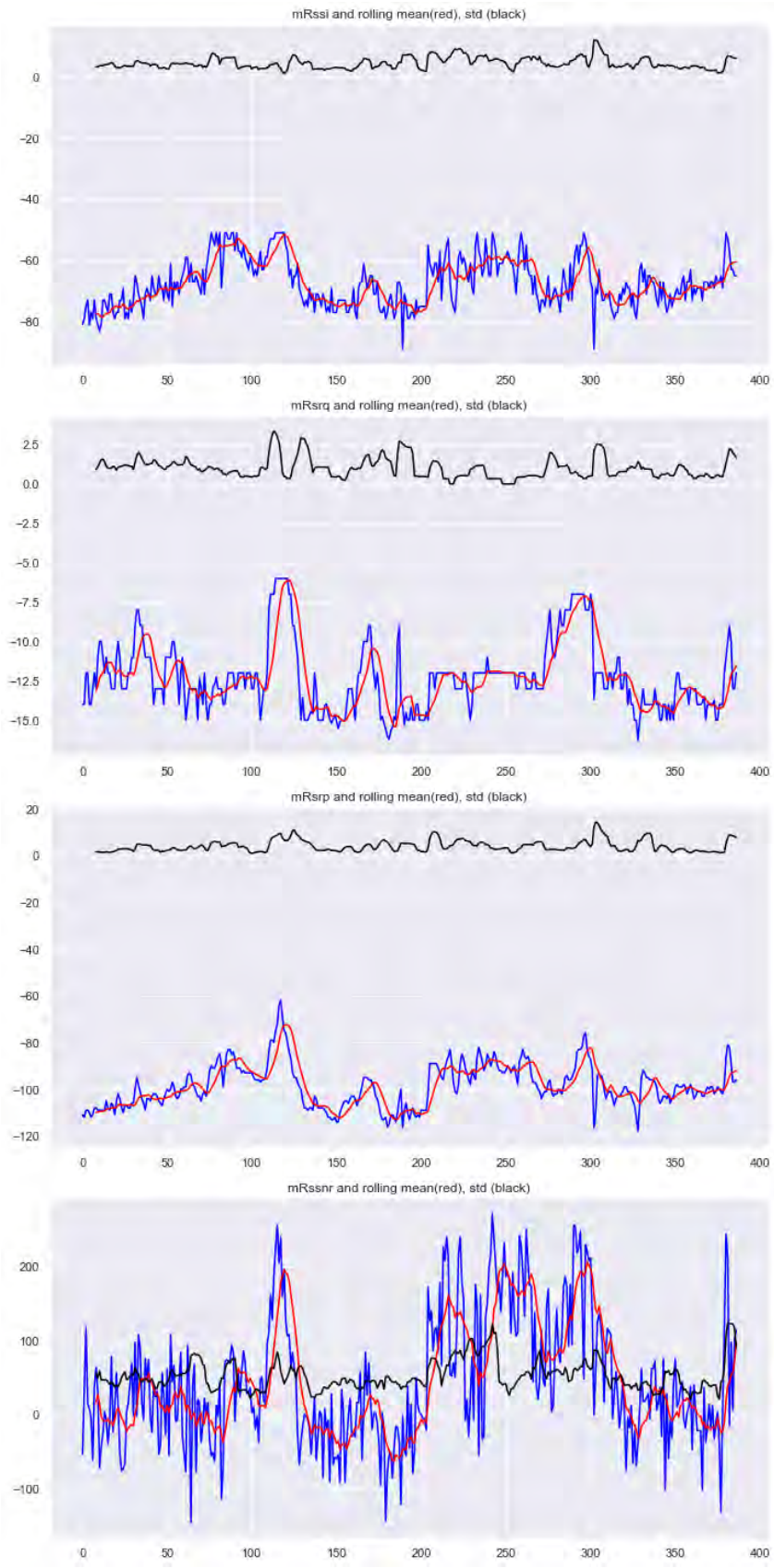

Figure 18: Rolling mean and std plots for the same QoS measurement excerpt

5.3: Mathematical Terminology

To efficiently study and describe a sample consisting of one or multiple features that are indexed by their instance on the temporal dimension, we will arrive at the definition of many key concepts that aim first at explaining, then creating a theory-backed structure for the process that is assumed to be generating the data at hand and finally expanding across time, which is our target.

5.3.0: Fundamentals

- **Mean** of a sample X is the sum of sample values (Xi) divided by their count:
$$mean(X) = \frac{\sum_{i=1}^{n} Xi}{n}$$
- **Variance** of a sample X is the average of the squared differences from the sample mean: var(x)=$\sum_{i=1}^{n} \frac{(Xi-mean(X))^2}{n}$. It measures the average degree to which each number is different form the sample mean. Variance is higher in value when a sample is comprised of a wider range of numbers, while lower in value when the sample exhibit a narrower variety of subpopulations. It gives more weight to outlier values because of the square exponent in its calculation
- **Standard Deviation** measures the degree of dispersion in a set of numbers, compared to the sample mean and is calculated by taking the square root of the variance i.e. std(X) =$\sqrt{var(X)}$
- If we have a random sample that consists of n data values represented by $X_1, X_2, \dots, X_N$ and comes from a population with a mean of μ(X) and standard deviation of σ(X), the sample mean mY provides a good estimate of the *population mean*, E(X). The sampling distribution of this statistic is derived from the **Central Limit Theorem** (CLT), which states that under very general conditions, the sample mean has an approximate normal distribution with mean E(X) and standard deviation σ(X)/ √n. There is no need for the population itself to be normal, however, the more symmetric the distribution of the population, the better is the normal approximation for the sampling distribution of the sample mean. This approximation tends to be better the larger the sample size n.
- **Covariance** measures how two variables in a common sample fluctuate together: cov(X,Y) =$\sum_{i=1}^{n} \frac{(Xi-mean(X))*(Yi-mean(Y))}{n-1}$
- **Autocovariance** refers to the covariance between time points of a single process.
- **Stationarity,** in its most basic form, refers to the statistical properties of Time Series data e.g. mean and variance being independent of time.
- **Second-Order structure** is used to refer to the variance and covariance in time.

5.3.1: Time Series Data & Autocorrelation

A **time series** is a sequence of data points, measured at successive time instances/ time points. Time series analysis comprises of methods attempting to uncover the underlying context of the data, by means of finding a Data Generating Process (DGP) with the aim of either explaining or forecasting/predicting. Forecasting using time series analysis is related to using a model of choice to forecast future events, making use of historic behaviour. A time series model generally reflects the fact that observations that come close in the temporal axis are more correlated than observations further apart. There exist 3 major families of time series models of practical importance, utilized extensively in Econometrics literature, namely, *autoregressive* (AR), *integrated* (I) and *moving average* (MA) models (not covered in this thesis). A wild range of variations exist, with the most common being *autoregressive moving average* (ARMA) and *autoregressive integrated moving average* (ARIMA) models [35].

Autocorrelation measures the correlation of time series data with past instances of itself (lags) i.e. the correlation of X_i *and* X_{i+k}. This is exploited by the use of the *Autocorrelation Function* (ACF), used for detecting non-randomness and to identify an appropriate time series model for the data under examination. Given a series of measurements $X_1, X_2, \dots, X_N$, the *lag* k ACF is defined as [36]:

$$r_k = \frac{\sum_{i=1}^{N-k}(X_i - \bar{X})(X_{i+k} - \bar{X})}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

Common practice in time series analysis is the visual inspection of the ACF of the time series under consideration. Some common behaviours of ACF plots can be noted below:

- If the ACF decays slowly over increasing time lags, it is usually a sign of *non-stationarity* of the time series, depicting dependence on the time axis i.e. the sample's statistical properties depend on time, providing moderate predictability if modelled properly.
- If the ACF exponentially decays towards zero for increasing time lags, it is a clear sign of stationarity of data, a very convenient case, since most time series models require the assumption of *stationarity*.
- If the ACF fluctuates around the lag axis with fixed frequency, it is usually a sign of *seasonality* i.e. repeated patterns/behaviour across time that have been unaccounted for.
- If the ACF after the 0^{th} lag is inside the confidence interval regions i.e. the series poses statistically insignificant autocorrelation at all lags, the time series at hand is deemed a *White Noise* process of zero mean and finite variance.
- AR models have sinusoidal ACFs that converge to 0.

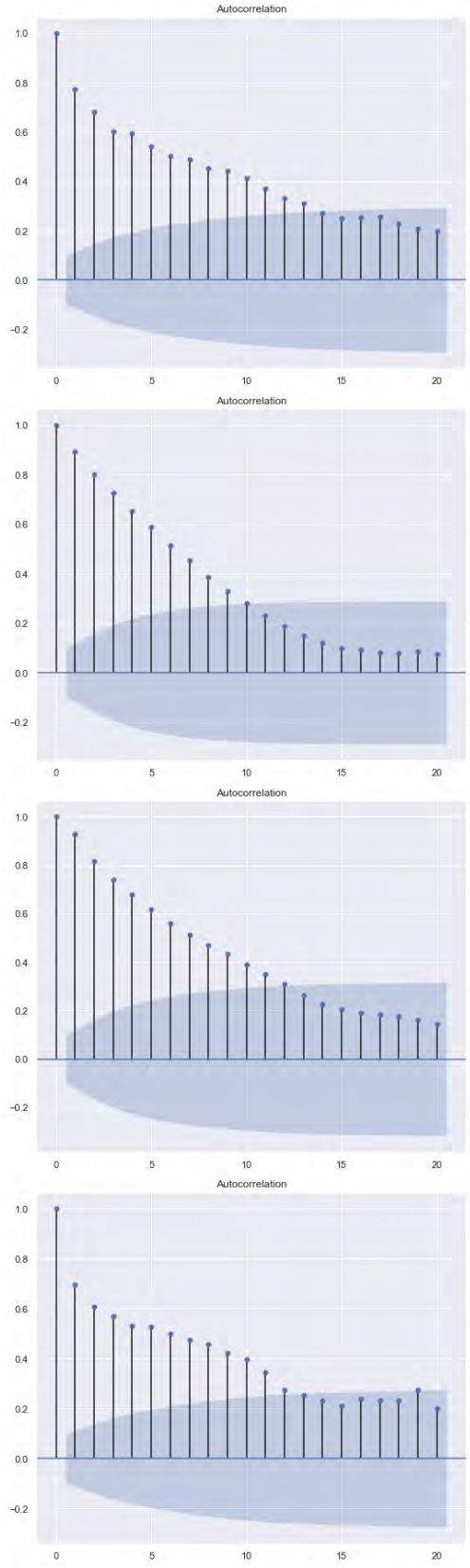

Figure 19: Example of measured QoS ACF plots

The above figure shows the ACF plots for an example excerpt of 3GPP QoS measurements concerning a single mobile user in the network. Top to bottom, are the plots for mRssi, mRsrq, mRsrp and mRssnr, showing moderate autocorrelation, thus predictability. From the ACF plot one can visually decide on the utilized lag length, meaning how many past instances of the data are useful to be used as regressors with the aim of inferencing or forecasting the time series in hand via appropriate modelling. For example, in the illustrated excerpt of data, all variables seem to benefit from at least 10 utilized lags i.e. all variables seem to have 10 lags of statistically significant information.

In real-world deployments, model specification through visual inspection is infeasible, so a way to automate various procedures such as the ACF is of paramount importance. Nonetheless, our final framework can optionally generate ACF plots, among other plot categories, created per sample window and labelled with the timestamp of the utilized sample's last observation, for further visual inference by post-process examination.

5.3.2: On Non-Stationarity

A **Stationary Process** is a random process that generates data points with a constant mean, variance and covariance i.e. data fluctuate around the deduced mean value with a finite variance. Examples of stationary time series are White Noise, MA(3), AR(1), etc.

White Noise (WN) is a stationary process of uncorrelated random variables with zero mean and constant variance. WN is a model of an absolutely chaotic process of uncorrelated observations, immediately forgetting its past.

Covariance stationarity involves the stability of the sample mean and covariance structure, across time i.e. the mean is stable and the covariance depends on the related points' separation in time but not the time instance itself.

Strict Stationarity refers to the *mean-stationarity* (stability of the mean structure), *variance stationarity* (finite and constant variance) and covariance stationarity aspects [37].

If non-stable behaviour, in terms of mean and variance, is observed, then the time series at hand is deemed as **non-stationary** and special care needs to take place. Non-stationarity can be statistically inferred, upon evidence on the existence of one or multiple traits listed below:

- Structural Breaks
- Seasonality
- Deterministic and/or stochastic trending components
- Human-related factors

A time series may exhibit one or multiple **structural breaks**. Structural breakpoint detection is usually carried out first i.e. performing structural change tests to infer on the existence of one or multiple breakpoints in the series' structure across time e.g. a sudden change in the level or variance. We can perceive a structural break as what the sudden drop in stock market time series during the 1930's is to Econometricians. In our context, structural breaks can translate to sudden changes in channel quality related to factors such as blockage, contention, etc. Usual practice is to discard breakpoint(s)-prior data in context of stabilizing behaviour across time. For our framework, we employ the Bai & Perron approach to structural breakpoint detection, as offered by R's *strucchange* package, which is

able to simultaneously estimate a fixed number of breakpoints in mean and/or variance, utilizing a low-order linear vector autoregressive process, with the goal of minimizing an Information Criterion (IC) like the Bayesian IC (BIC), alongside the RSS (Residual Sum of Squares) from the regressions on the hypothesized sample segments. Upon completion, the procedure reports possible breakpoints in each metric's structure across time and quantifies on the importance of those breakpoints.

Seasonality patterns are usually next, able to render our time series as Seasonal. Single or multiple seasonality can efficiently be eliminated to stationarity by applying *seasonal differencing*. Seasonal models can be more easily detected, modeled and predicted than data which lack seasonality, as in our case.

Trending components of deterministic and/or stochastic nature, rendering our time series as **Trend Stationary/Difference Stationary.** *Trend Stationarity* refers to the cases where stationarity is achieved by performing an operation to remove the trend component of our time series data. Trends are generally of deterministic nature e.g. time/linear trends, quadratic trends etc. and can be considered a rarity in our data's nature. *Difference Stationarity* refers to the cases where applying one or multiple times the *first differencing* operator on our time series, achieves stationarity. The forecasts of Trend Stationary models tend to the trend line, while those of Difference Stationary models are not *mean-reverting* i.e. at any point the process begins anew.

Non-stationarity can also be induced by commonly practiced operations, performed erroneously by a data engineer, such as concatenation of non-temporally contiguous datasets or aggregation and distortion of available data. A series of transformations has to be applied to the non-stationary data, to grant a stable behaviour that will result in effective model specification, estimation and prediction.

5.3.3: Data Scaling Transformations

Most data can benefit from one or a chained sequence of transformations [32]. Reasons for doing so include but are not limited to:

- Improving assumptions of *normality and linearity* and potentially eliminating non-linearities
- Improving assumptions of *homogeneity of variance*
- Making units comparable when having different ranges/scales
- Reducing the effect of total quantity and focusing on relative quantities
- Equalizing or altering the relative importance of common and rare subpopulations in the posed data
- Emphasizing informative subpopulations at the expense of uninformative ones

Monotonic Transformations are applied to each element of a data matrix, independent of the other elements. They are "monotonic" because they change the data point's value without changing their *rank.* Examples include but are not limited to:

1. Log Transformation ($y = \log(x)$): Very popular in the field of econometrics, especially for count data. Can generally handle well non-zero and non-negative data. Ad-hoc

solutions can be found for incorporating zero and/or negative values, such as adding a quantity before applying log. This is reported to destroy the relative orders of magnitude i.e. not treating positive, negative and zero values symmetrically. Thus, in our case this transformation is unusable.

2. Power Transformations ($y = x^p$): Another popular family of transformations, where the value of the exponents defines the transformation's behaviour e.g. p=0 gives presence/absence, p=0.5 is a square root transform, etc. The smaller the exponent, the higher compression it applies to higher values.
3. Square Root Transformation ($y = \sqrt{x}$): Similar with the log transform, can handle zero but is less drastic in behaviour.
4. Arcsine Transformation ($y = \left(\frac{2}{\pi}\right) * \sin^{-1} x$): This transformation scales the arcsine (in radians) to [0,1]. Input must already be in this range, otherwise one should first relativize.
5. Cubic Root Transformation ($y = \sqrt[3]{x}$): Is essentially a power transformation, compressing outliers and stabilizing the variance, all while mapping any real number to a real number i.e. can handle either negative, zero or positive values. Cubic Root has a problem with its derivative at zero, if used as an activation function.
6. Inverse Hyperbolic Sine Transformation ($y = \sinh^{-1} x$): IHS is an interesting family of transformations which also stabilizes variance, reduces the effect of outliers and also has a real to real number mapping. It can be equivalently written as $y = \log(x + \sqrt[2]{x^2 + 1})$. Its ability to be used as an activation function for all real values is noted.
7. First Differences Transformation ($y = \Delta X_t = X_t - X_{t-1}$) is a very powerful transformation used for mean and variance stabilization by effectively removing deterministic trends and seasonal patterns, tackling stochastic trend components, otherwise known as *Unit Roots,* while being able to remove potential structural breaks, giving rise to the problem of *overdifferencing*. One of its fundamental uses is for achieving stationarity of time series data. Combining with a log-based scaling transformation, non-linearities can also be smoothened out.
8. Relativizations adjust matrix elements by a row/column standard (e.g. subtracting the min/max/mean in a column/row-wise manner). Indeed, whenever there is a time series Y_t at hand to be studied and modelled, we can subtract the sample mean, redirecting our study to $Y_t - \mu$, so that the process is expressed in deviations from its mean, which has zero mean by construction, without loss of generality [37].

Other options may be *probabilistic transformations* e.g. Beal's Smoothing, deleting rare subpopulations i.e. outliers, combining entities.

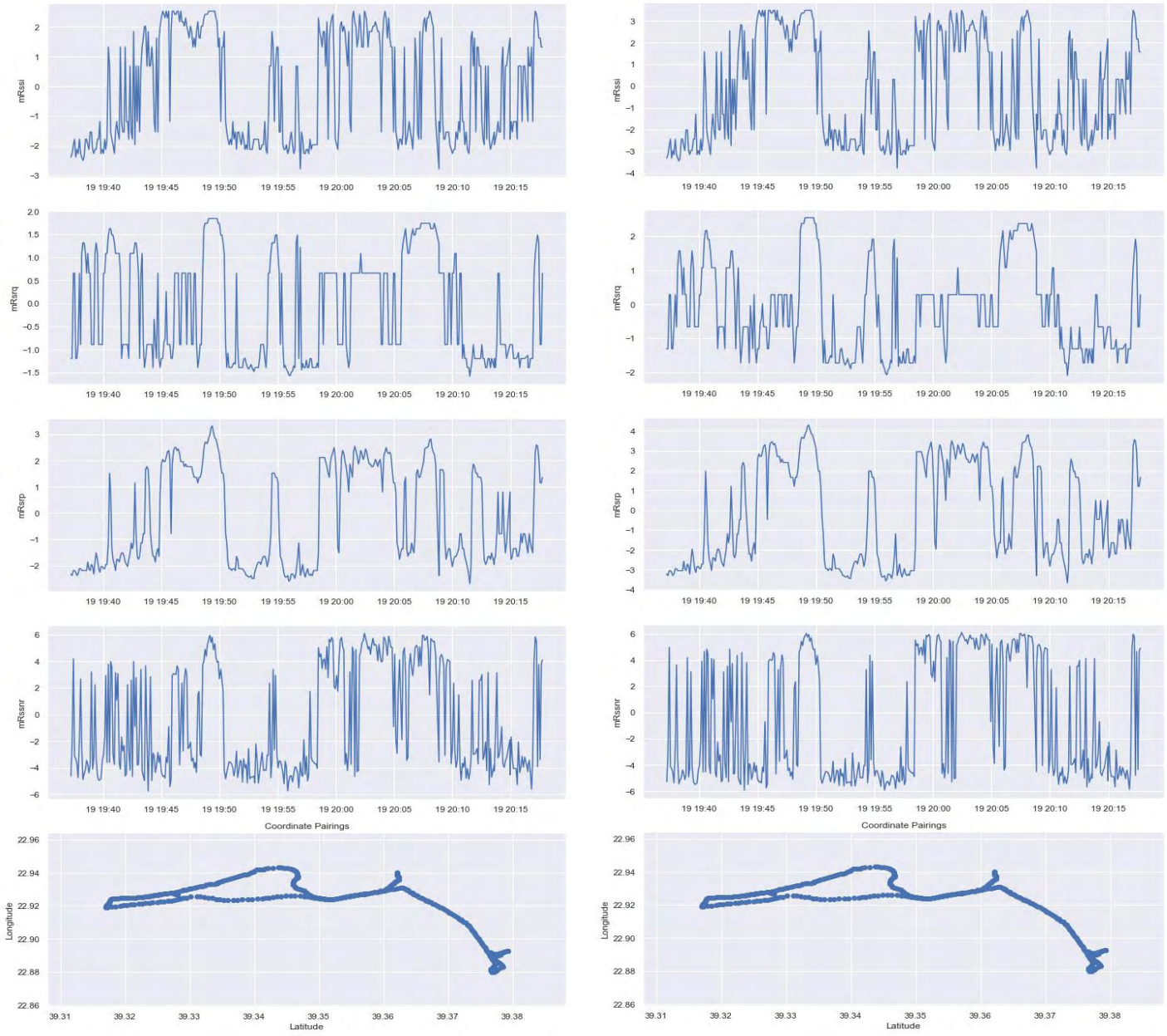

Figure 20: Cubic Root (Left) and IHS (Right) transformations

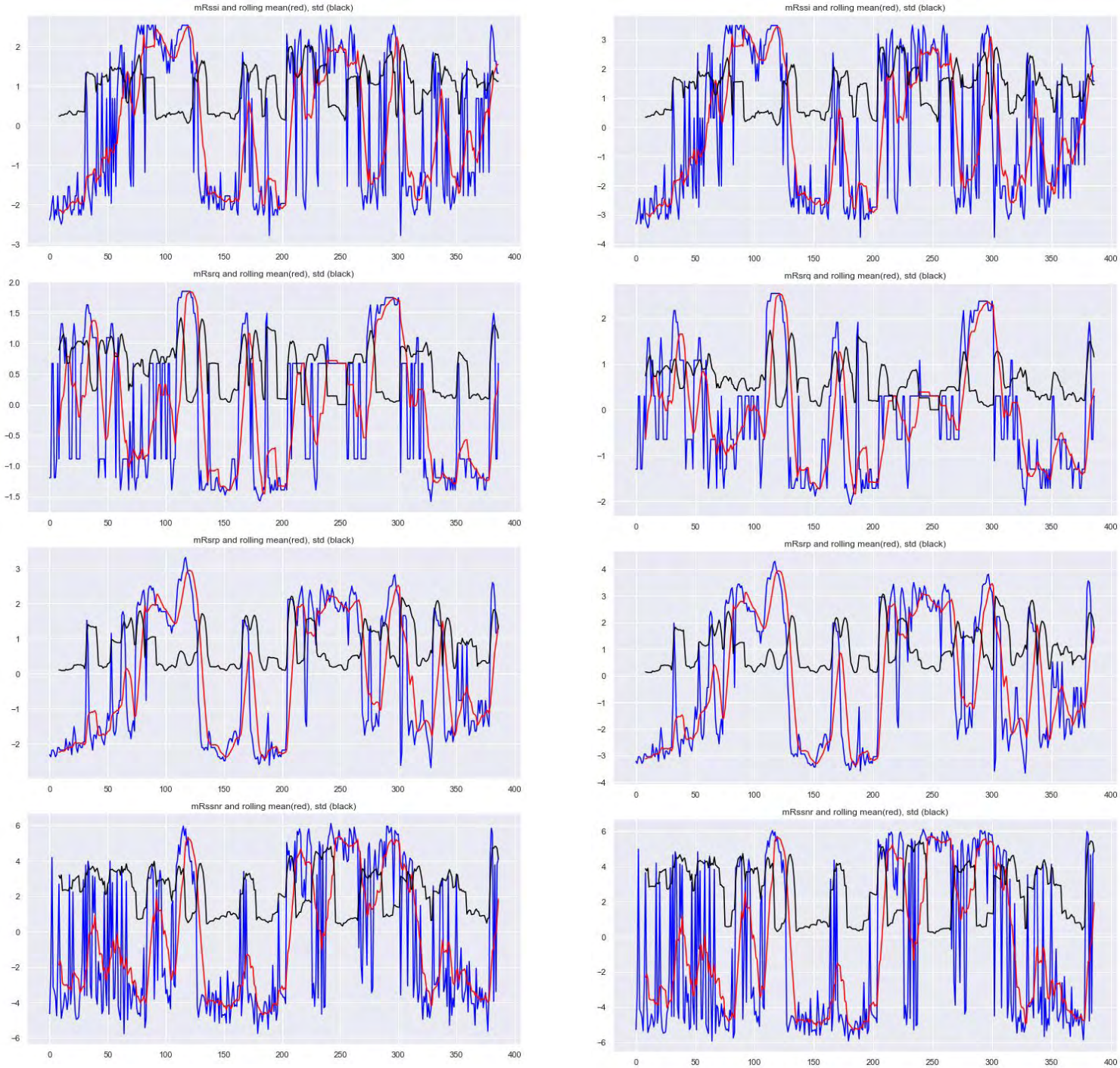

Figure 21: Rolling mean and std plots for Cubic Root/ IHS –transformed series

As can be observed from the figures above, each time series is first zero-centered, while the mean and variance structures are stabilized well by the respected scaling transformations applied, keeping the relative magnitudes intact and moderately smoothing outlier subpopulations, while rendering our variables as effectively comparable. A visual comparison with the raw, untransformed data plots from Section 5.2.2 is encouraged for confirmation of the above statements.

5.3.4: AR(p) processes

Autoregressive processes contain **lags** (past values) as predictors. This means that a value in time is estimated by OLS-deduced weighting of the process' lag values. The number of lags contained in the model formula is denoted by *p.* A first order (p=1) autoregressive process, denoted by *AR(1),* is expressed as:

$$Y_t = \varphi Y_{t-1} + \varepsilon_t \,, \quad \text{where} \quad \varepsilon_t \sim WN(0, \sigma^2) \ (1)$$

Equivalently: $(1 - \varphi L)Y_t = \varepsilon_t \longrightarrow Y_t = \frac{1}{(1-\varphi L)} \varepsilon_t \,,$

where L denotes the Lag operator:

$$LY_t = Y_{t-1},\ L^2 Y_t = Y_{t-2} \text{ , etc.}$$

The lag operator usually forms a polynomial of degree m:

$$B(L) = \beta_0 + \beta_1 L + \beta_2 L^2 + \cdots + \beta_m L^m$$

An AR(1) is stationary if |φ| < 1. To elaborate, the condition for covariance stationarity is that the root $z_1 = \frac{1}{\varphi}$ of the lag operator polynomial is greater than 1 in absolute value, while generalizing on the AR(p) case, all polynomial roots are greater than 1 in absolute value i.e. lie outside the unit circle. An interpretation of the latter, is that if $\sum_{i=1}^{p} \varphi_i \geq 1$, the process is not stationary.

AR(p) processes possess the ability to uncover/explain more highly persistent dynamics in a system, relative to the p parameter value, while *MA(q)* processes have very short memory, regardless of the moving average length q parameter. A generalization of the AR(p) model and properties for multiple variables is called ***Vector AutoRegression or VAR(p)*** and will be used in our study.

The p parameter can be deducted from various Information Criteria (IC) such as Akaike (AIC), Bayesian (BIC), Schwartz (SC), Hannan-Quinn (HQ), etc. Each IC defines a function that can be efficiently minimized by means of Maximum Likelihood Estimation (MLE). Reportedly, AIC favors larger orders, while BIC results in more parsimonious models. Larger lag length leads to larger number of regressors in our model equation, having a direct impact on the variance of the produced model and predictions.

5.3.5: Unit Roots

Unit roots are essentially stochastic trending components that damage our model stability. Using a model that is dynamically unstable for forecasting purposes invalidates our study and consequently our forecasts. A common case where this can happen, is mistakenly identifying a Difference Stationary Process for a Trend Stationary one. OLS estimation of the model weights is inconsistent, since the ε_t error is no longer a stationary White Noise process i.e. the regression *residuals (remainders)* are autocorellated, thus non-stationary.

Based on equation (1), if the root $z = \frac{1}{\varphi}$ of an inverse characteristic equation $1 - \varphi z = 0$, is very close or equal to 1, we say that this AR(1) process has a unit root and is thus, difference stationary. An AR(1) process with a unit root is called a *random walk.*

Generalizing for an AR(p) process $Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{t-p}$, a unit root is present if *at least one* root of the inverse characteristic equation $1 - \varphi_1 z - \varphi_2 z^2 + \cdots + \varphi_p z^p$ is equal to 1.

Unit Root and Stationarity testing is a well-covered topic in Econometrics, where approaches of Dickey-Fuller (DF/ADF), Kwiatkowski-Phillips-Schmidt-Shin (KPSS), Phillips-Perron (PP), Zivot-Andrews (ZA), etc. are popular in usage, each with its tradeoffs. In our framework we apply a combination of tests, presented in later sections.

5.3.6: On Integration

A time series variable X_t is called ***integrated of order d*** or **I(d)** if stochastic trends persist after d-1 times of first differencing, but are eliminated with the dth differencing operation, finally rendering our process as stationary. Recalling the differencing operator of the previous section i.e. $\Delta X_t = X_t - X_{t-1}$ and to elaborate on the above: X_t is I(d) if $\Delta^d X_t$ is stationary, while $\Delta^{d-1} X_t$ still has a stochastic trend, or is, equivalently, Difference Stationary. A k-dimensional vector of time series or an MTS of k variables is I(d) if *at least one* component is I(d).

5.3.7: On Spurious Regression

A striking amount of observable time series, not only in the area of Economics, but generally in real-world settings, are non-stationary and require careful study. Granger & Newbold in [39], were the first to point out that performing regression with non-stationary variables can produce *spurious* or nonsensical results, dating back to 1974. To elaborate, if all regressors are I(1) and not cointegrated, we have spurious results, observing high R^2 scores and a low Durbin-Watson statistic of autocorrelation. That is, on model fitting, the used variables seemingly regress well on each other but on performing prediction, invalid results are produced. This owes to the residuals of the regression being correlated, thus exhibiting non-stationary behaviour, leading to the inconsistency of the OLS estimation, which is based on the stationarity of the regression residuals. Spurious regression applies to Trend Stationary and Difference Stationary processes as well [40]. Phillips proved in 1986 that the problem of Spurious Regression cannot be remedied by larger sample sizes, when clearly an orthological approach is to identify non-stationarity as carefully as possible [41].

5.3.8: On Cointegration

An exception to the problem of Spurious Regression, on performing regression with non-stationary variables is when the model eliminates the stochastic trends that are present, to produce stationary residuals from the regression. That is the *cointegration* phenomenon, where two or more non-stationary variables share common stochastic trend components. The cointegration phenomenon can be found with an intuitive example of a drunk person walking with his/her dog, where the two seemingly take random paths but are not allowed to be separated further than the dog's leash. Regressing one non-stationary variable to another non-stationary variable, when the two have a cointegrating relationship, actually produces *superconsistent* OLS estimates i.e. having a higher convergence rate that the stationary case. To formulate [42], let $Y_t = (y_{1t}, y_{2t}, \dots, y_{nt})'$ denote a (n x 1) vector of I(1) time series. Y_t is *cointegrated* if there exists a *linear combination* of its components that is stationary:

$$\beta' Y_t = \beta_1 y_{1t} + \cdots + \beta_n y_{nt} \sim I(0)$$

If the (n x 1) Y_t is cointegrated, there can be r linearly independent cointegrating vectors, ranging from 0 to n exclusive, forming a basis for the cointegrating vector space. Since most cointegrating vectors are not unique, a normalization procedure often takes place. In our framework, the Johansen approach to cointegration testing is employed, covered in next sections in a more detailed manner, while we note on the applicability of alternative approaches.

5.3.9: Granger Causality

One of the most common practices in data analysis is quantifying the similarity in the behaviour of two or more variables. Commonly, correlation matrices and distance-based comparison are carried out. Correlation, though, is not indicative of *causality*. True causality refers to a process whose behaviour enables the behaviour of another. However, since true causality is impractical to be observed, we arrive at the definition of **Granger Causality**, a popular concept in the field of Econometrics. As proposed by Clive Granger in 1969, causality involves a time series variable Granger-causing another, if the latter can be better predicted if it combines the history of the causing variable in the regression, contrary to using only its own historic behaviour. Alternatively, if variable X contains valuable historic information that leads to reduced residuals when regressed on Y with the aim of predicting Y, we say that *X Granger-causes Y*.

Granger Causality tests involve hypothesis testing having as null that coefficients for the causing variable are all zero. From such tests we can conclude that X G-causes Y, Y G-causes X or both. Granger Causality tests can be performed only on the levels of the times series variables, while utilizing a VAR model, i.e. entered variables must be I(0).

5.4: Tests for Statistical Inferencing

5.4.0: Hypothesis Testing

Hypothesis Tests extract information from sample data to infer on whether a starting condition is likely to be rejected, usually concerning specific parameter values of a model or a sample while measuring inference uncertainty. Specifically, such tests explicitly specify:

- The *null hypothesis*, denoted as H_0 i.e. the starting axiom/condition.
- The *alternative hypothesis*, H_1, which is employed on rejecting H_0.
- The utilized *test statistic* and its distribution under H_0.
- Selection of the *significance level a,* so as the *rejection region* can be determined.
- Calculation of the test statistic from the sample data.
- Conclusions based on the produced test statistic and the rejection region.

The **null hypothesis** is presumed to be true until the data provides sufficient evidence that it is false, always relative to the significance level a. In Hypothesis testing, we can either *reject* or *fail to* reject H_0. If we fail to reject, it does not mean that the null is necessarily true i.e. the procedure neither can choose on the truth of H_0 and H_1, nor which is most likely to be true. It can only assess whether sufficient information exists to reject the null.

After stating our H_0, we next test it against the **alternative hypothesis.** For example, if our null is H_0: $\beta_i = c$, we can specify the H_1 in three possible ways:

- H_1: $\beta_i > c$ i.e. accepting that $\beta_i > c$.
- H_1: $\beta_i < c$ i.e. on rejecting H_0, we accept that $\beta_i < c$.
- H_1: $\beta_i \neq c$ i.e. on rejecting H_0, we accept that β_i is either greater or smaller than c.

The **test statistic** is estimated under H_0 i.e. assuming that it is true. Under the null, the distribution of the statistic is known. Based on the produced value of the test statistic, we decide on whether to reject or fail to reject H_0. The most widely used distribution for Hypothesis testing is the *t Distribution*, derived from the standard normal and chi-squared distributions. Based on the t distribution, a *t-statistic* is calculated, whereupon finding that the null is not true, the t-statistic does not have a t-distribution with N-2 *df*, but can be explained by another.

A **rejection region** is comprised of values that possess low probability in being true when the null is true and depends on the H_1 specification. If the estimated test statistic value lies between the rejection region bounds, something unlikely to occur under the null, then it is unlikely that the null holds, leading to our rejection. The size of the rejection regions is relative to the **level of significance α**, merely specifying the sensitivity of the test i.e. a probability of the unlikely event, usually 0.05, 0.01, or a more relaxed 0.1. For example, a value of α = 0.05 implies that the null hypothesis is rejected 5% of the time when it is in fact true. To decide on rejecting or failing to reject, we compare the calculated test statistic to the *critical value* provided. If the test statistic is *greater in magnitude* (in absolute value) than the critical value at the chosen cut-off point α, we can reject the null. Critical values are dependent on the test statistic and the significance level.

Another approach to the one presented above is the **p-value** method, where a suitable value is decided prior to the test and controls how the test rejects. Choosing a p-value is completely analogous to choosing a significance level in the previous approach. For example, we decide either to reject the null hypothesis if the test statistic exceeds the critical value for a significance level (e.g. α=0.05) or reject the null if the p-value is less than its predetermined value of 0.05. In our framework, we mainly employ the first approach, with a few exceptions of employing the p-value approach in cases where the critical values are not available by the utilized software.

5.4.1: Testing for Unit Roots and Stationarity

Autoregressive Unit Root Tests attempt to infer on whether the time series at hand is Trend Stationary (φ<1) or Difference Stationary (φ=1). The null hypothesis is typically the existence of a unit root in the AR process i.e. its difference-stationarity, with the alternative of stationarity. They are called unit root tests, since the null hypothesizes that the autoregressive polynomial of z_t, $\varphi(z) = (1 - \varphi z) = 0$ has a root that is equal to unity.

Stationarity tests typically start with the null hypothesis of Trend Stationarity, with an alternative of existence of Unit Root in the AR polynomial. First differencing on trend stationarity produces a unit moving average root in the ARMA representation e.g. the first-differenced ARMA(1,1) representation is $\Delta z_t = \varphi \Delta z_{t-1} + \varepsilon_t + \theta \varepsilon_{t-1}$ with θ=-1. This is called *overdifferencing.* Typically, stationarity tests look for a unit moving average root in Δz_t.

Through comparison of test-specific t-statistics and simulated critical values (that are dependent on the defined distribution of each test and the inclusion of deterministic terms), we can infer on the question posed on a significance level desired. Options for inclusion of deterministic terms in the regression usually include: a) a constant value b) a linear/time trend c) a quadratic trend d) combination of a constant and a trend e) none. Traditionally, it is important to visualize the series beforehand, to conclude on the options posed above, observing the series' behaviour e.g. if we observe a linear trending behaviour, we opt for b), etc. In our case, the relativization and scaling transformations applied prior to Unit Root/Stationarity Testing and the nature of our QoS data can ease on the options to be employed, leading to the configuration of the tests according to option e) or at most a).

5.4.1.1: On Lag Length Selection & ADF Unit Root Test

Said & Dickey in 1984 built on the traditional Dickey-Fuller Test which employed a procedure applicable only for AR(1) processes with WN errors. Their approach, namely, Augmented Dickey-Fuller or ADF Test was able to capture more complex dynamics in a time series through accommodation of general ARMA(p,q) models with unknown p, q parameters. The null hypothesis is that the given time series y_t is I(1), implying existence of a unit root, with an alternative hypothesis that it is I(0), assuming ARMA dynamics. The ADF test evolves around the test regression:

$$y_t = \beta' D_t + \varphi y_{t-1} + \sum_{j=1}^{p} \psi_j \Delta y_{t-j} + \varepsilon_t \quad (2)$$

In the equation above, D_t is a vector containing the deterministic terms defined. The p parameter is used to approximate the ARMA structure of the errors, with a value that ensures no serial correlation.

An important test hyperparameter is the lag length specification, denoted by p. If it is smaller than appropriate, the remaining serial correlation in the residuals biases the test results. If p is larger than necessary, the power of the test is impaired. Methodologies for lag length selection include:

- Data-driven methods e.g. Ng and Perron [57].
- Information Criteria minimization e.g. AIC, BIC, SC, HQ, etc.
- Fixed value, traditionally determined by visual inspection

In our implementation, the lag selection phase is performed in a multiprocessing manner, employing various approaches. They are based on VAR model selection techniques, having R's VARSelect method as a basis, with the aim of minimizing the *Akaike Information Criterion (AIC),* either in a brute-force manner which favors larger values, or in a serial correlation-tested manner which employs the *Breusch-Godfrey* approach and typically leads to smaller lag lengths that produce well-formed residuals. The maximum lag length to be investigated is defined in a data-driven manner i.e. is a function of the sample size.

AIC is used extensively by modern research as a means to quantify on relative model adequacy, where a lower score is indicative of a preferable model, with the aim of predicting/forecasting future values. Its equation is: $AIC = -2 * \ln(\hat{L}) + 2 * k$, where $\hat{L}$ is the likelihood value and k is the number of estimated model parameters.

ADF Tests tend to over reject the null of unit root for finite small samples and coupling with Stationarity Tests that use as null the alternative hypothesis of the unit root tests, is recommended for a more robust approach to stationarity.

5.4.1.2: KPSS Trend Stationarity Test

Stationarity tests, as mentioned, define as the null hypothesis that our series is I(0), with an alternative of I(1). The KPSS procedure [44] assumes that a time series can be decomposed into the sum of a deterministic trend $\beta' D_t$, a random walk μ_t and a stationary error u_t:

$$y_t = \beta' D_t + \mu_t + u_t$$
$$\mu_t = \mu_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN(0, {\sigma_\varepsilon}^2)$$

, where D_t is the vector of deterministic components, as in the ADF case and u_t is I(0). KPSS test statistic is the Lagrange Multiplier for testing ${\sigma_\varepsilon}^2$=0 against the alternative of ${\sigma_\varepsilon}^2 > 0$.

Our approach to finding the I(d) of our already detrended and scaled time series, includes iteratively first differencing our series until the two tests dictate otherwise, for a decided significance level α (e.g. 0.05 or 0.01), while permitting no deterministic terms, other than a constant at most. As deterministic trend components in typical QoS measurements of mobile users can be viewed as rarities, on employing the approach above we essentially use the KPSS test as a safety measure to overdifferencing. The *urca R* package provides the basis for our I(d) Inferencing Phase.

5.4.2: Cointegration Testing

5.4.2.1: Cointegration in the VAR framework

The Granger representation theorem links cointegration to Error-Correction models (ECM). Soren Johansen's work linked cointegration and ECM in a VAR framework, where the levels VAR(p) for the (n x 1) vector Y_t:

$$Y_t = \Phi D_t + \Pi_1 Y_{t-1} + \cdots + \Pi_p Y_{t-p} + \varepsilon_t$$
$$t = 1, \dots, T$$
$$D_t = \text{deterministic terms (e.g. constants, linear/quadratic trends, etc.)}$$

- Is stable if $\det(I_n - \Pi_1 z - \cdots - \Pi_p z^p) = 0$ i.e. all of its roots z lie outside the complex unit circle.
- If there are roots on the unit circle (unit roots) then some or all the variables in our vector are I(1), while cointegrating relationships are possible.
- If Y_t is cointegrated then the usual VAR representation is not so efficient in explaining the cointegrating relations. Derived directly from the VAR specification, a **Vector Error Correction** (VEC) model is more suitable and is defined as:

$$\Delta Y_t = \Phi D_t + \Pi Y_{t-1} + \Gamma_1 \Delta Y_{t-1} + \cdots + \Gamma_{p-1} \Delta Y_{t-p+1} + \varepsilon_t$$
$$\Pi = \Pi_1 + \cdots + \Pi_p - I_n$$
$$\Gamma_k = -\sum_{j=k+1}^{p} \Pi_j \,, k = 1, \dots, p-1$$

- In the VEC framework, ΔY_t and its lags are I(0) i.e. only applicable to I(1) variables (which makes the most sense, since we can't talk of cointegration for I(0) variables).

- If the VAR(p) process has unit roots (z=1) then Π is a singular matrix i.e. has *reduced rank*, denoted as rank(Π)= r < n.
- If rank of Π equals to zero, then it is implied that Π is a zero matrix and that $Y_t \sim I(1)$ and not cointegrated. VEC is thus reduced to a VAR(p-1) in first differences:

$$\Delta Y_t = \Phi D_t + \Gamma_1 \Delta Y_{t-1} + \cdots + \Gamma_{p-1} \Delta Y_{t-p+1} + \varepsilon_t$$

- If rank of Π equals to r < n, then it is implied that $Y_t$is I(1), having r cointegrating relations i.e. r linearly independent cointegrating vectors. Π can be written as a product of two (n x r) matrices with rank r, named α and β', respectively. The VEC is altered as:

$$\Delta Y_t = \Phi D_t + a\beta' Y_{t-1} + \Gamma_1 \Delta Y_{t-1} + \cdots + \Gamma_{p-1} \Delta Y_{t-p+1} + \varepsilon_t$$

  where $\beta' Y_{t-1}$is I(0) i.e. stationary, due to the β' component being a matrix of cointegrating vectors.
- If Π has full rank i.e. rank is equal to r = n, then we have the special case of a **Cointegrated VAR** that is stable and stationary.

5.4.2.2: Johansen's Approach to Cointegration

Johansen's approach can be divided in 4 steps:

1. Specify and estimate a VAR(p) model for Y_t.
2. Construct likelihood ratio tests for the rank of Π to determine the number of cointegrating vectors.
3. If necessary, impose normalization and identify any restrictions to the cointegrating vectors.
4. Given the normalized cointegrating vectors estimate the resulting cointegrated VECM by means of maximum likelihood.

Johansen's Trace Statistic formulates a Hypothesis Testing Procedure:

$$H_0(r) : r = r_0 \,, \qquad H_1(r_0) : r > r_0$$

The LR statistic, called the *trace statistic*, is given by:

$$LR_{trace}(r_0) = -T \sum_{i=r_0+1}^{n} \ln(1 - \hat{\lambda}_i)$$

- If rank(Π) equals to r_0, then $\hat{\lambda}_{r_0+1}, \ldots, \hat{\lambda}_n$ should all approximate zero as values and the LR trace statistic should be small, owing to $\ln(1 - \hat{\lambda}_i) \approx 0$ for i > r_0.
- If rank(Π) is greater than r_0, then some of the eigenvalues will be nonzero and less than 1, while the LR trace statistic should be large, owing to $\ln(1 - \hat{\lambda}_i) << 0$ for some i > r_0.

Our approach to Cointegration is based on the ca.jo module, offered by the R *urca* package. One important thing to note is that Johansen's approach is only applicable to variables that are Integrated of the same order and the order is I(1). For mixed orders of Integration but not I(2), the ARDL Bounds Test to cointegration [52] is an interesting

approach. The latter approach consists of essentially an AR(p) univariate process which contains a variable set of lag lengths for each participating regressor, denoted by q. The distribution of the test statistic permits a mixture of orders of integration for our right hand side of the equation i.e. our independent variables, while our dependent variable has to be I(1) and not suffering from structural breaks. One can employ the ARDL R package [55] which offers an instantaneous and well-formed approach.

5.4.3: Causality Testing

5.4.3.1: Toda-Yamamoto approach to Granger Causality

What commonly follows the stationarity aspect is the one of Cointegration Testing for pairs of integrated variables. However, one can be more interested in the causality of the variables under inspection, where Granger Causality Tests are usually performed, as prior mentioned. If the pair of time series under inspection is stationary, then the Granger procedure can be performed, yielding consistent results. Testing for Granger-causality using F-statistics when one or both time series are non-stationary can lead to *spurious causality* [53]. We continue using a popular approach to Granger Causality, proposed in [38]. The procedure is based on:

1. Determining the maximum order of Integration present in our multiple time series, denoted by *m.*
2. Fitting a VAR model in the variables' levels i.e. without having applied differencing.
3. Determining the lag length (p). The VAR model utilized is a VAR(p), with respect to the assumption of no serial correlation in the residuals.
4. Forming an augmented VAR, denoted by VAR(p+m), increasing the lag length by m, as deduced in step 1.
5. Carrying out a Wald test for the first p variables only with p degrees of freedom.

The VAR model is augmented by adding lags equal in number to the max order of integration, as deduced from the previous Unit Root/Stationarity tests, when on the Wald residual correlation test, the coefficients of the variance-covariance matrix are included up until the originally selected lags. The serial test employed in step 3 is the Breusch-Godfrey test, with the null hypothesis that residual errors are not serially correlated. A higher p-value shall be preferable. An example of how this works: We find that Variable 1 is useful in forecasting Variable 2, but not the other way around. That is because the p-value of the first Wald Test (H0: Variable 2 does not Granger-Cause Variable 1) was e.g. 0.75 so it can't be rejected at a desired confidence interval of, say, 10%. While on the second Wald Test (H0: Variable 1 does not Granger-Cause Variable 2) the p-value was e.g. 9.46e-5 leading to our easily rejecting the Null at 10%.

There is also the concept of Confounding variables that is important, but not tackled in this thesis. That is when higher level dynamics between variables exist. For example: A causes B and C, while B causes C. While we might find the causations as A->B, A->C, B->C, we could go for a filtering of those relations, basically as A->B, A->C. An extension of this framework to remedy for this effect is set as future work.

5.5: On Neural Networks

Neural Networks are certain algorithms that, inspired and loosely modeled after the human brain, are efficient in pattern recognition, concerning data in various formats such as text, numerical values, image, sound, which are usually exerted by modern real-world deployments. All data destined as input must be converted into a suitable format such as an array or a vector, or generally, a *tensor.* This family of algorithms has proven useful in problems related to text classification and image recognition.

A neural network is consisted of one or multiple *layers*, which are in turn comprised of one or multiple nodes/*units*. The node is the quintessential computational part in a neural network, which is stimulated on input, altering the received data by using weights for denoting significance of input. The produced values are subsequently summed and passed through an *Activation Function*, essentially a scaling transformation with all the benefits that were noted in Section 5.3.3.

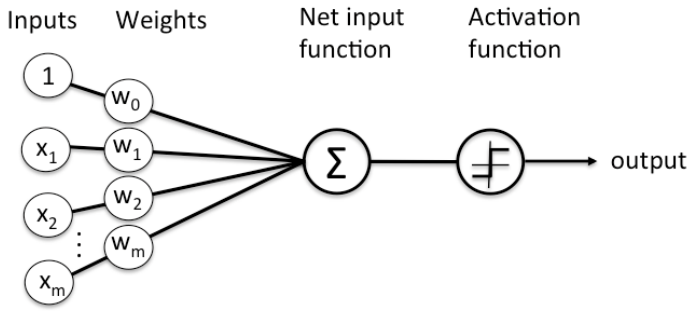

Figure 22: Neural node diagram

A node layer is a stack of nodes whose output is redirected appropriately to the next layer's input. Namely, we can have input, hidden and output layers. The output layer usually squishes the calculation to interpretable dimensions. A neural network that employs multiple layers of nodes, contrary to legacy one-input/one-output layer Perceptrons, is qualified as a *Deep Neural Network* (DNN) [45].

Neural Networks utilize optimization functions that perform weight estimation/re-estimation, according to the error induced, through the *gradient descent* method i.e. mapping the network's error as a function of the utilized weights in the form of a derivative dE/dw, quantifying on the relative differences between the weight and error values. By aiming at minimizing the error produced by the weighting of the nodes in all layers, they are prone to *overfitting* i.e. they fit the training datasets exceptionally well, at the cost of not being able to generalize well to non-typical input data behaviours, encountered in test/validation datasets. Gradient-based *optimization algorithms* can also be stuck to local minima due to non-convex optimization and this can be tackled by *Keras' Early Stopping mode* which stops the batch training on a fixed number of failed consecutive tries in further minimization of the produced loss. Keeping a relatively small number of units per layer, even as small as the number of regressors, can introduce well-generalizing traits to the NN predictions' behaviour.

Neural Networks are reportedly adequate in handling large amounts of data that result in the estimation of needed parameters, while being able to tackle non-linear

behaviours and relationships. Deep learning networks are also reportedly better when fed with large input amounts and are thus considered data-hungry. The more the input, the more calculations they have to make, the more associations they have to evaluate and consequently, the higher time complexity is introduced. We should remind that from the Plasticity vs Stability scope, excessive data-feeding will lead to lessened errors i.e. increased predictive power in terms of finding correctly a mean behaviour but due to the robustness itself, sudden impulses are likely to be considered unimportant small-scale variations, hence lacking in plasticity i.e. agility in behaviour, all while requiring lengthy training times.

5.5.1: Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) is a family of Neural Networks proposed in the 1980's for modelling and predicting time series. Its structure resembles legacy multi-layer Perceptrons, with the addition of inter-hidden unit connections, associated with a time delay. Through these connections the model can pick up and retain historic input data behaviours and uncovering temporal correlations between time instances with a large separation on the time axis [46].

RNNs can pose difficulties in their proper training, despite their powerful nature, mainly owing to the effects of the *vanishing gradient* and *exploding gradient*, introduced in [47]. To clarify, the latter denotes the large increase in the gradient's norm during the training phase. Such events are caused by the explosion of the long term components, growing exponential orders of magnitude more than their short-term counterparts. The former problem is essentially the opposite case, where the long-term components tend exponentially fast to 0, damaging the training process and the RNN's ability to infer on associations/correlations between distant time lags. The NNs hidden state at time t, denoted by h_t , is actually an update over the previous hidden state h_{t-1} , passed through a smooth and bounded function e.g. a logistic sigmoid ($S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$) or a hyperbolic tangent function ($\tanh z = \frac{\sinh z}{\cosh z} = \frac{e^z-e^{-z}}{e^z+e^{-z}} = \frac{e^{2z}-1}{e^{2z}+1}$). The traditional recurrent hidden state update follows the equation $h_t = g(Wx_t + Uh_{t-1})$ and the flow of the updates is depicted below on an unrolled RNN.

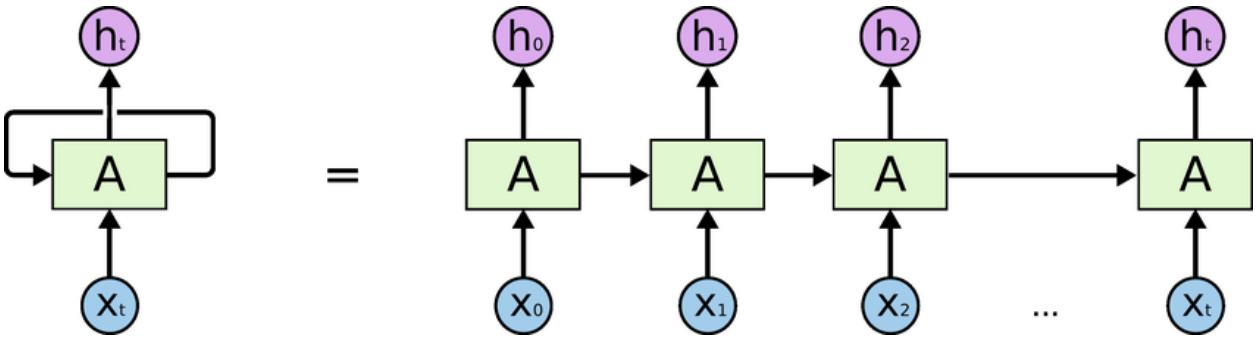

Figure 23: A simple RNN structure

5.5.2: LSTM RNNs

The pathogenies of gradient-based RNNs lead to the design of more sophisticated activation functions, consisting of affine transformation followed by a simple element-wise nonlinearity by using gating units [48]. A first attempt was the creation of a recurrent unit, called **Long Short-Term Memory (LSTM)** unit [51]. LSTMs proved useful for cases related to language modelling and traffic flow prediction.

An LSTM cell typically possesses four gates: the *input, input modulation*, *forget* and *output* gates. The input gate is the point of entry for new data and the output gate is for handing-off the result to the next calculation level/layer. The forget gate utilizes a sigmoid function to decide on whether to discard/forget certain information, resulting in a systematic time lag deducing procedure. The memory cell input gate receives as input the result of the LSTM cell of the last iteration.

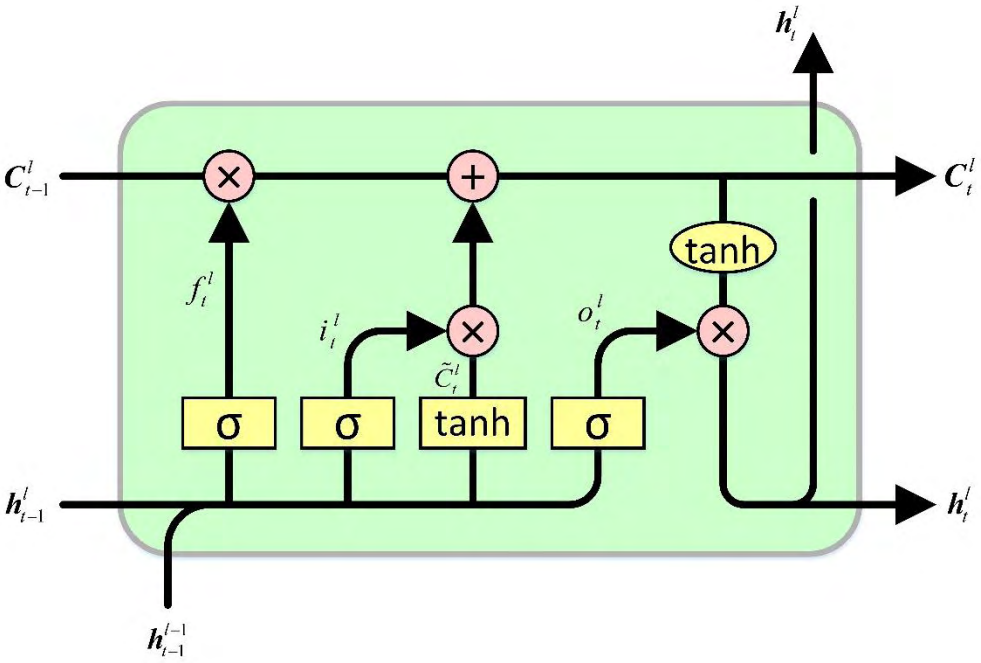

Figure 24: An LSTM cell/block

On being presented with an input of a *univariate* time series, denoted with X and comprised of instances in time $X_t, X_{t-1}, \dots, X_{t-T}$, while the hidden state of the memory cells is $H = h_1, h_2, \dots, h_n$ and the output time series as $Y = (y_1, y_2, \dots, y_n)$, LSTM RNNs perform computations accordingly:

$$h_t = H(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$
$$p_t = W_{hy}y_{t-1} + b_y$$

One can observe that the calculation of the hidden state at time t relies on the weighted (W matrices are weight matrices) input data and previous hidden state at time t-1, summing along the *bias vectors* b_h.

5.5.3: GRU RNNs

The **Gated Recurrent Unit (GRU)** is a relatively recent specification that is based on the LSTM architecture [50], tackling computational issues around the latter. They have reportedly outperformed LSTMs in Natural Language Processing (NLP) problems, while introducing leveraged ease in implementation. A typical GRU cell consists of a *reset gate* denoted by *r* and an *update gate*, which is a combination of the LSTM forget and update gates i.e. GRUs employ 2 gates while LSTM utilize 3, merging the cell state and memory state amongst other changes.

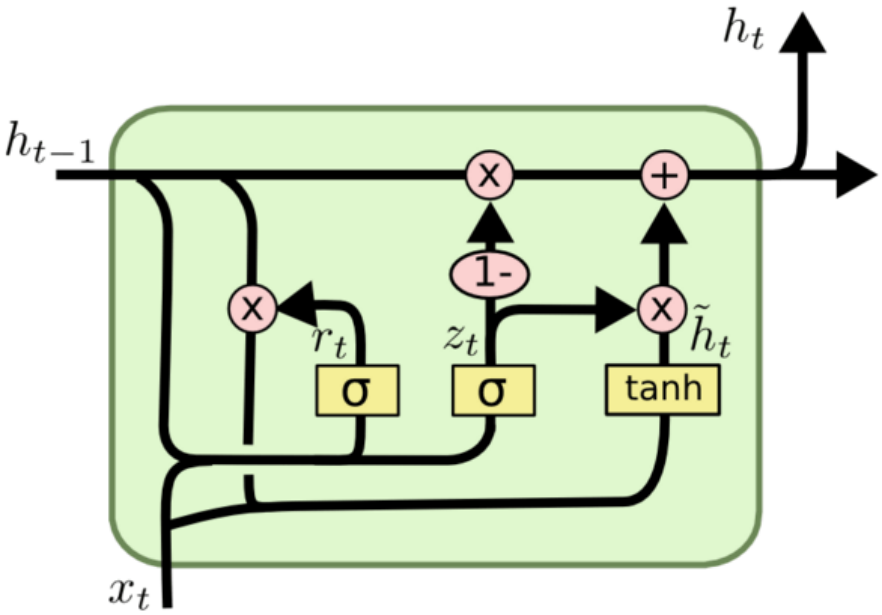

Figure 25: A GRU cell/block

In our approach, we allow for a formation of a DRNN of multiple lightweight LSTM and GRU layers, employing tanh(x) or/and $\sinh^{-1}(x)$ activation functions. The utilized sample is divided in two parts, the training set and the validation set, with the latter consisting of the last few observations of the current sample window, being useful for fine-tuning the estimated weights, prior to forecasting.

5.5.4: Univariate & Multivariate Autoregressive Time Forecasting

Univariate autoregressive modelling for multi-step time forecasting results in using the ML/NN model's own univariate point forecasts for moving forward in the forecast horizon. As the forecast horizon competes with the lag length in value, one can realize that the larger the horizon we choose, the larger that the error can potentially be, with respect to the utilized lag length. What is more, we are restricted in using as predictors only the considered variable's history, with the only possibility of adding easily-predicted deterministic terms or exogenous variables, missing out on the benefits of the multivariate

approach. One remedy to this, is generating time forecasts for the regressors/independent variables with the VAR family of models that was presented before, essentially feeding our DRNN with VAR-generated synthetic data so as the forecasts can be reweighted accordingly. An alternative approach that can be employed in our framework but won't be showcased in this thesis, would be feeding the DRNN with the residuals of the VAR model fitting, with the final forecasts being a combination of VAR time forecasts and DRNN residual forecasts. The latter is introduced in [56], where the authors claim that the aviation data time series at hand can be modelled in two phases, a linear and a non-linear one. They assume a stable and stationary VAR(1) for linear modelling and forecasting one or multiple steps ahead in time, to provide for the OLS-deduced expected behaviour, to later combine it with the gradient descent-deduced expected behaviour of the forecasting errors i.e. training and forecasting with RNN variations on the VAR regression residuals. The resulted formulation, called R2N2, is reported by the authors to exhibit leveraged performance in comparison to each separate model's performance on aviation data. While the authors choose to not carry out common model specification techniques, they claim that their formulation can remedy for that. Generally, ignoring an elaborate VAR model specification procedure can allow for cases of spurious regression and an impaired foundation for our final forecasts. Furthermore, the authors do not impose restrictions on the utilized lag length for the DRNN and proceed with training.

Our framework permits various approaches for generating the final forecasts and the relative performance of some will be later showcased. Through extensive hypothesis testing and inferencing operations, we are able to generate a theory-backed VAR/VEC model to fit and forecast our multiple time series, that can solely be used to forecast multiple steps ahead in time and/or be combined with a DRNN, to assess the mixture's potential. Noted also is the approach of employing a higher-order VAR model specification that is likely to overfit our sample data but can side-step on the problem of serial correlation of the residuals, while typically possessing a more volatile behaviour. The training of the DRNN model is performed in a lightweight manner, employing Early Stopping on non-convergent behaviour, while employing Nesterov-augmented ADAM optimization (NADAM) in a try to tackle RNNs' tendency to get stuck in local minima. The input data are transformed and formulated based on the results of the procedures that were presented earlier, effectively reducing the number of regressors, towards a more lightweight and informative cycle of operations. Loss function options vary in behaviour, as will be shown later. In our examples, we employ either *MAE* or *LogCosh* loss functions. Multiple DRNN formulations are able to be trained in parallel, each one including the information that can statistically provide increased forecasting power, in support of forecasting each QoS measure of interest, employing a multivariate approach. Moving forward in the forecast horizon uses the VAR-generated synthetic data that is prior generated. Parallel computations are performed by assigning each computationally-intensive operation to a Python's *multiprocessing* Process, whenever possible, to successfully harvest the processing power of a multi-core system, effectively cutting down on our operations cycle time.

5.5.5: On Forecasting Error Metrics

To compare our implemented approaches, we first need to define the appropriate error metrics that will quantify the error accompanying our time forecasts. We can typically come across 3 types of evaluation measures:

Scale-dependent metrics build around the raw difference between the observed and predicted quantities. They are characterized by ease in computation and examples include the *Mean Squared Error (MSE)*, defined as: $\frac{1}{n}\sum_{i=1}^{n}(actual_i - forecasted_i)^2$ and the *Mean Absolute Error (MAE)*, defined as: $\frac{1}{n}\sum_{i=1}^{n}|actual_i - forecasted_i|$. Such methods come with certain disadvantages, involving their dependency on the data scale i.e. comparing the MSE or MAE of multiple time series is meaningless [58]. Another option is the *Geometric Mean Absolute Error* (GMAE), defined as: $\sqrt[n]{\prod_{i=1}^{n}|e_t|}$, i.e. the geometric mean of the absolute errors, though flawed in that any error term that is estimated as zero, will render the GMAE as zero, usually for cases where we can have zero forecasts.

Percentage-error metrics, in the general form of $p_t = 100\ \frac{e_t}{X_t}$, are used for measuring the error for each period, expressed as a percentage of the same period's observed demand. Being scale-independent, they allow for comparison of forecasting methods across multiple data series. A popular example is the *Mean Absolute Percentage Error (MAPE),* defined as $\frac{1}{n}\sum_{i=1}^{n}|p_t|$, which, again, suffers in the zero case, yielding undefined values.

Scale-free error metrics introduce a more versatile approach, while not suffering from problems for certain ranges of the forecasts. A popular example is the *Mean Absolute Scaled Error (MASE)*, defined as:

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Actual_t - Forecasted_t|}{\frac{1}{n-1}\sum_{j=1}^{n}|Y_j - Y_{j-1}|}$$

,where the denominator calculates the MAE for the naïve method and the metric yields a number that is less than 1 if our forecasts analogously outperform the naïve method or higher than 1 if the naïve method provides a more accurate forecast.

For our comparative analysis of implemented multivariate forecasting approaches, we choose to employ either of the following metrics: *Mean-Relative Squared Error* (MRSE) and *Relative Error* (RE). MRSE and RE metrics can be formulated accordingly:

$$MRSE = \frac{\sqrt{\sum_{i=1}^{n}\sum_{t=1}^{T}(Actual_{it} - \widehat{Forecasted}_{it})^2}}{\sqrt{\sum_{i=1}^{n}\sum_{t=1}^{T}(Actual_{it} - mean(Actual_i))^2}}$$

$$RE = \frac{\sqrt{\sum_{i=1}^{n}\sum_{t=1}^{T}(Actual_{it} - \widehat{Forecasted}_{it})^2}}{\sqrt{\sum_{i=1}^{n}\sum_{t=1}^{T}(Actual_{it})^2}}$$

, where the *it* subscript denotes the i-th variable at time t. The first metric quantifies on the quality of the model's prediction as compared to naively predicting each variable's mean, while the second provides a measure of the error that is relative to the scaling of our data [56].

5.6: On Spatial Analysis & Interpolation

5.6.0: Fundamentals

First and **Second-order effects** are important as concepts in Spatial Analysis and underline its basic principles, studied by density-based measurements. The former denote how observations vary from neighbourhood to neighbourhood due to changes in the underlying statistical properties, commonly employing Kernel density estimations, which assess the 1st order structure of the underlying process. The latter denote how observations vary from neighbourhood to neighbourhood, due to interaction effects between observations, commonly employing Average Nearest Neighbours (ANN) and K-functions to assess the 2nd order structure of the DGP.

On a sample of measurements dispersed in the spatial domain, *interpolation* refers to producing a map of measurements, containing values at every location in the input grid of locations. Typically, for any unknown point, some form of weighted average of the surrounding values is performed, essentially creating a contiguous surface from a given set of points. The first rule of geography by Waldo Tobler in 1970 notes that *everything is related to everything else, but near things are more related than distant things,* comprising the basic premise behind spatial interpolation and the reason why nearer points are assigned greater weights than more distant ones.

Methods for interpolating in the spatial domain include *deterministic* and *geostatistical* methods. The former category includes algorithms such as Radial Basis Functions (RBF) and Inverse Distance Weighting (IDW), which uses mathematical functions to calculate the values at unknown locations based either on the degree of smoothing or the similarity to neighbouring points in space. The latter category uses both mathematical and statistical methods to predict values at all locations within a region and to provide uncertainty measures for the quality of the interpolation performed with respect to the spatial autocorrelation present. This category includes the Kriging family of unbiased estimators and its variations.

5.6.0.1: Spatial Autocorrelation

Spatial Autocorrelation refers to the existence of systematic spatial variation in a considered random variable. Where adjacent realizations are similar in value, the relationship is denoted with a positive value. On the contrary, if adjacent observations have contrasting values then this value is negative. Detecting spatial autocorrelation can prove beneficial in some cases, implying an unobserved underlying distribution of map values for investigating the spatial variations. Further, it can imply information redundancy and can pose implications in spatial data analysis procedures. Examples of spatial autocorrelation measures, specified in terms of covariances are *Moran's I* and *Geary's c.* The former is defined in the same manner with the time-domain ACF, incorporating the spatial weights matrix and is used to provide an explanation on how the values of the variable under examination are related to the locations measured:

$$I = \frac{n}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}}$$

5.6.0.2: On variography

Geostatistical methods introduce the covariance-distance relationship to interpolation models, so as the needed weights can be calculated. To elaborate, the distances between unknown points at which prediction is desired and points that contain measurement, in the vicinity of the former, are calculated using the value of the *covariogram* for those distances, to efficiently determine the neighbouring weights. The covariogram relates the magnitude of distance between points to their covariance i.e. expresses covariance as a function of distance.

In some cases, the covariogram estimation cannot be made directly, leading to the definition of the *variogram γ(h)*, attempting to explain the spatial structure of the data at hand. One can derive the covariogram from the variogram but not vice-versa. Fitting a variogram model usually comprises of selecting a model for fitting the generated variogram in the Least Squares sense. Common variogram model choices are exponential, spherical, Gaussian, etc. Variogram definition relies on the *second-order stationarity* of the regionalization function Z that explains the spatial structure of our data into large and small-scale variations and *intrinsic stationarity* which states that the variable Z(x+h) - Z(x) is stationary.

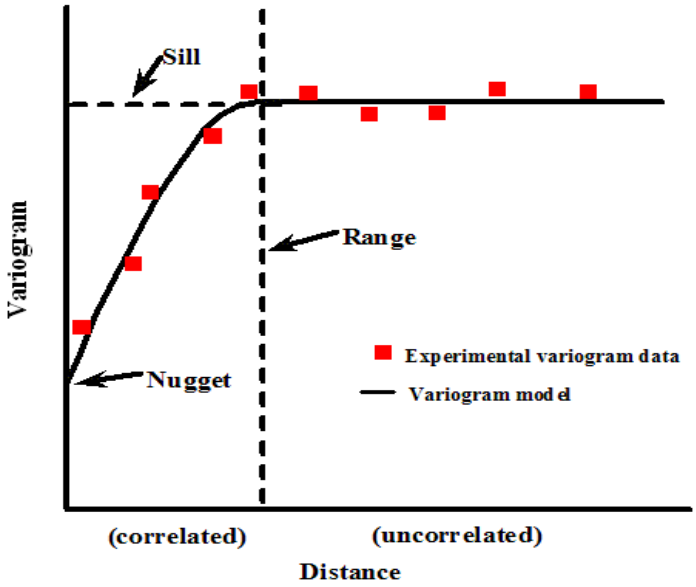

Figure 26: Experimental Variogram and Variogram Fitting

5.6.1: Inverse Distance Weighting (IDW)

Inverse distance weighting (IDW) interpolation explicitly makes the assumption that things that are close to one another are more alike than those that are further apart. To predict a value for any unmeasured location, IDW uses the measured values surrounding the prediction location. The measured values closest to the prediction location have more influence on the predicted value than those farther away. IDW assumes that each measured point has a local influence that diminishes with distance. It gives greater weights to points closest to the prediction location, and the weights diminish as a function of distance. Weights are proportional to the inverse of the distance (between the data point and the prediction location) raised to the power value p. As a result, as the distance increases, the weights decrease rapidly. The rate at which the weights decrease is dependent on the value of p. If p = 0, there is no decrease with distance, and because each weight λi is the same, the prediction will be the mean of all the data values in the search neighbourhood. As p increases, the weights for distant points decrease rapidly. If the p value is very high, only the immediate surrounding points will influence the prediction. In our approach, we first pass the data through a scaling transformation to improve the structure of the measured data that is to be interpolated. After the interpolation takes place, generalizing over the whole map/grid, we back-transform accordingly for results in the original scale.

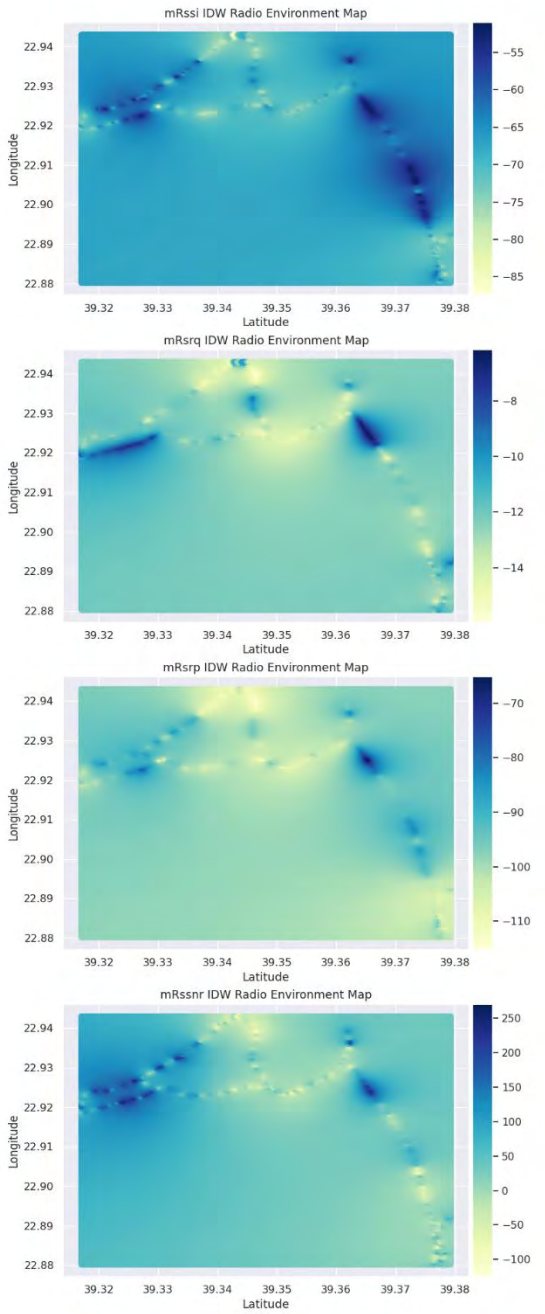

Figure 27: IDW Spatial Interpolation for measurements of a single Vehicular User

The above figure shows an IDW spatial interpolation over a grid created by 20000 points uniformly drawn from the bounding box of a single vehicular user's trajectory. One can observe that in the areas of measurement the map is more detailed, while in areas where there is lack of measurements for those locations, the metric is generalized close to the sample mean. For demonstration purposes, we used a power factor of 2, while cross-validation techniques for robustly deciding the power factor are noted. The resolution of the resulting interpolation grid and the number of sample points used, are hyperparameters of the accuracy vs calculation complexity tradeoff.

5.6.2: Trend Surface Analysis

As its name suggests, the scope of this analysis is to find the equation that best matches the attribute values of the data at hand. Trend Surface Analysis is often carried out using multiple linear regressions or polynomial functions of order 4 or 5 (e.g. $y = aX^4 + bX^3 + cX^2$) to model the to-be-predicted variable and a combination of the spatial coordinates. As an approach it is capable of incorporating all available data i.e. has a global nature. Approaches that are more focused on *local effects* can be more accurate in their predictions. Abnormal values and outliers reportedly damage this procedure, introducing a bad fit to the data. A suggested approach is to carry out the analysis without these values and to include them later in the interpolation, to validate the results.

Deterministic approaches to spatial interpolation such as IDW presented before, can be inadequate in interpolating if not used carefully, since uncertainty measures are absent, while not deriving a structure to explain the data at hand i.e. they are merely atheoretical.

That's the benefit of geostatistical approaches like **Kriging.** By capturing the spatial structure through the use of the previously mentioned variography methods, it provides a way of studying a field of measurements in a structured manner, providing uncertainty measures and preserving initial measurements into the interpolated map. Kriging methods suppose that a DGP can be divided into a deterministic trend component *m* that captures the large scale variations and an autocorrelated error *e(s)* i.e. the regression residuals as:

$$Z(s) = m + e(s)$$

, where Z(s) is the attribute value at location s, the trending component which is global for the field and the error term which is dependent on the location s. The variogram is computed on the residuals and depending on its resulted shape, the resulting interpolated map can be heavily smoothed, reducing the range of the predicted values, owing to a large *nugget* effect. Kriging is also reportedly sensitive to outliers and a suggested approach is estimating the variogram without these values, incorporating them in the subsequent interpolation procedure.

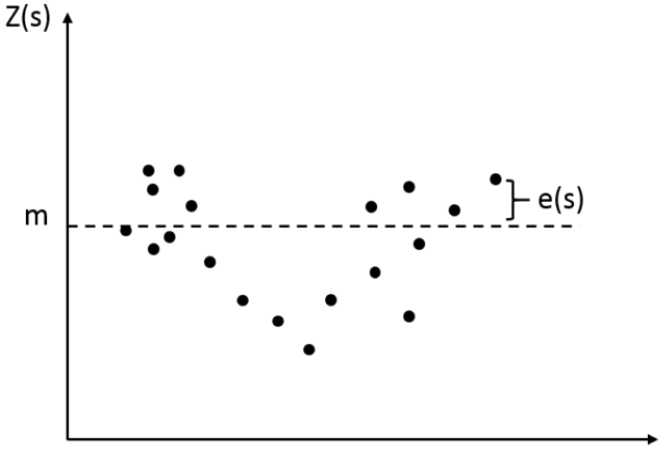

Figure 28: Simple Kriging's global approach

As mentioned beforehand, its global approach does not allow for local variations in the trend, being useful for cases where the samples behave in a very stable manner. It is thus considered inefficient for our problem statement, since QoS metrics can suffer from reflections and sudden fluctuations across space.

5.6.3: Ordinary Kriging

Ordinary Kriging assumes a constant trend only over the search neighbourhood for a given spatial point, allowing for local variations within a field. It is the most reported kriging approach due to its intuitive approach and efficient estimation. Its utilized function is:

$$Z(s) = m(s) + e(s)$$

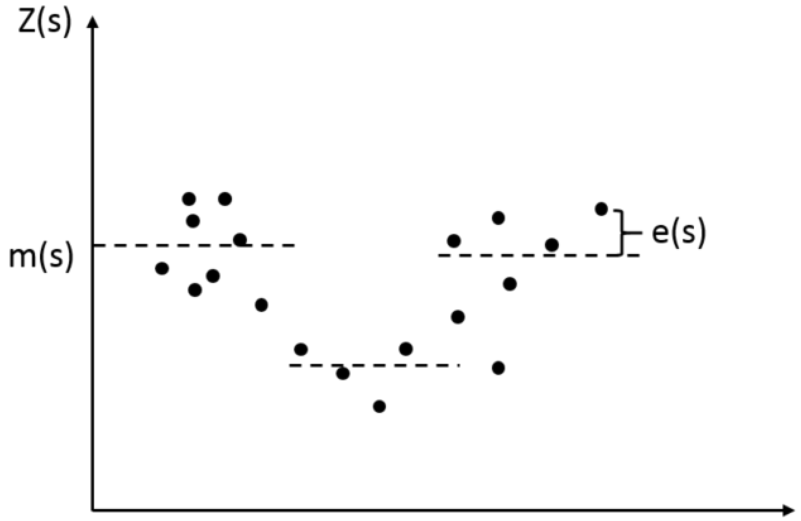

Figure 29: Ordinary Kriging's Local approach

The neighbourhood-local trend that is assumed, denoted with m(s), depends on the spatial location of the observation and is derived from the data measuring the intra-neighbourhood distances between pairs of sampled locations, pointing out the need for a careful definition of the extent of each neighbourhood.

In our implementation, the covariance in a metric between pairs of spatial locations, plotted as points along the distance/separation axis (in meters) i.e. the variogram, passes a covariance model selection procedure, where the variogram is fitted with a multitude of covariance models e.g. exponential, spherical, Matern and M.Stein-optimized Matern, as offered by the *autoKrige* method from R's *automap* package. The latter model seems to yield better results in some cases, being superior to exponential, spherical and Gaussian model fits. Ordinary Kriging on the received power alone can be adequate in terms of prediction error, compared to the case of modelling & removing the max signal path loss first, in the cases where the Base Station's location is known. In cases where transmitters can be multiple and their locations are not known, Kriging on the received QoS conditions is practical & recommended for building Radio Environment Maps (REMs) [33].

Borrowing conceptually from the crowdsourcing concept that is proposed in [34], we group measurements from similar users in a bounding box for near time instances, aiding in leveraged quality of the input data to the Ordinary Kriging procedure and in interpretable real-time results. Ordinary Kriging can reportedly result in non-prohibiting errors, even with relatively small sample sizes, while uncertainty measures are provided by R alongside the predictions, to offer an in-depth and detailed solution compared to the simplistic and atheoretical nature of the IDW procedure.

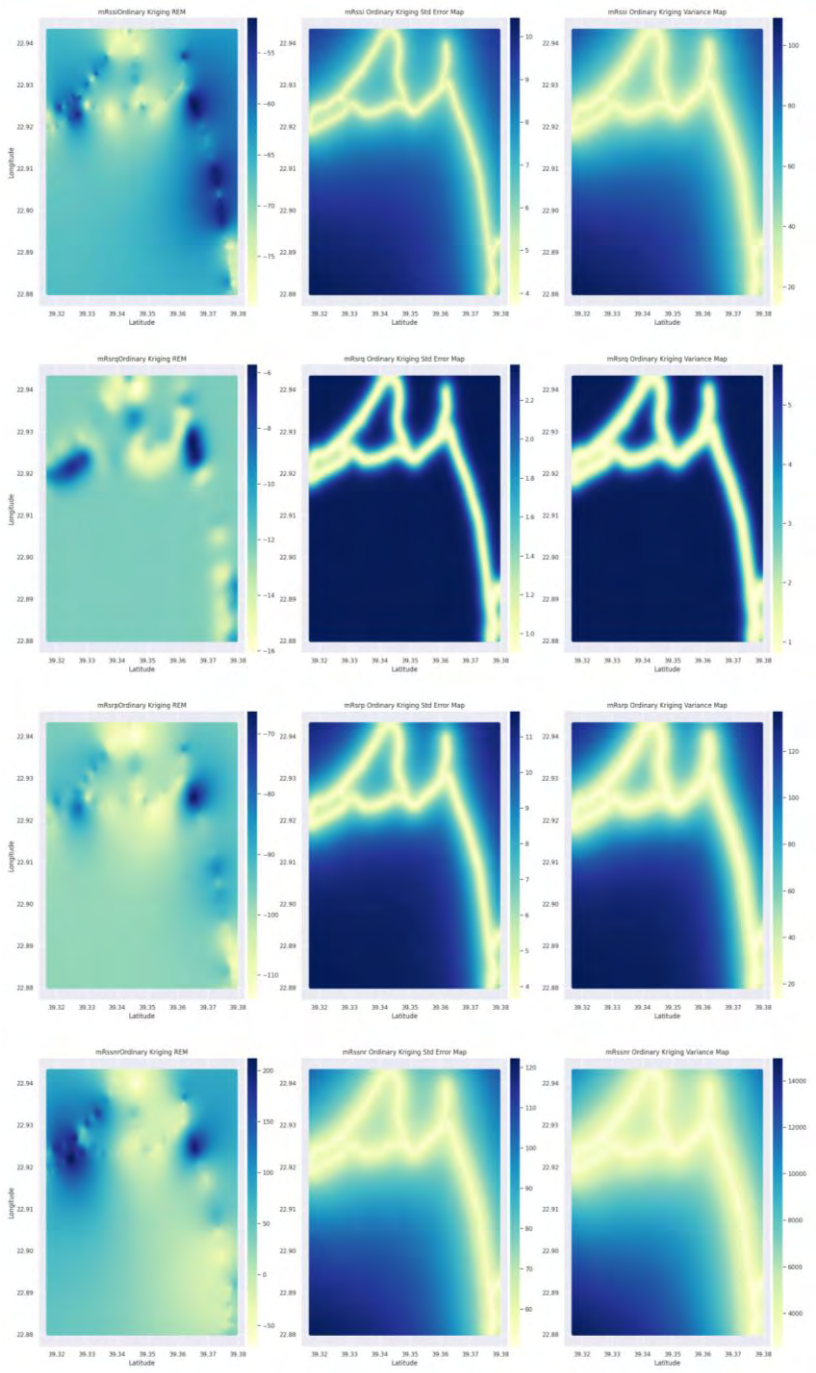

Figure 30: Ordinary Kriging on a single Vehicular User

The above plot was generated by forming a spatial grid as described in the IDW section and shares a common problem when single-user data are not dispersed well over the grid, resulting in predictions close to the sample mean, for sub-areas of no measurements. As can be observed, the reported standard errors are well-behaved, with the exception of the mRssnr metric. Kriging is orders of magnitude heavier computationally than IDW, but shares the latter's hyperparameters of sample size and spatial grid resolution.

5.7: Flow Chart of Operations

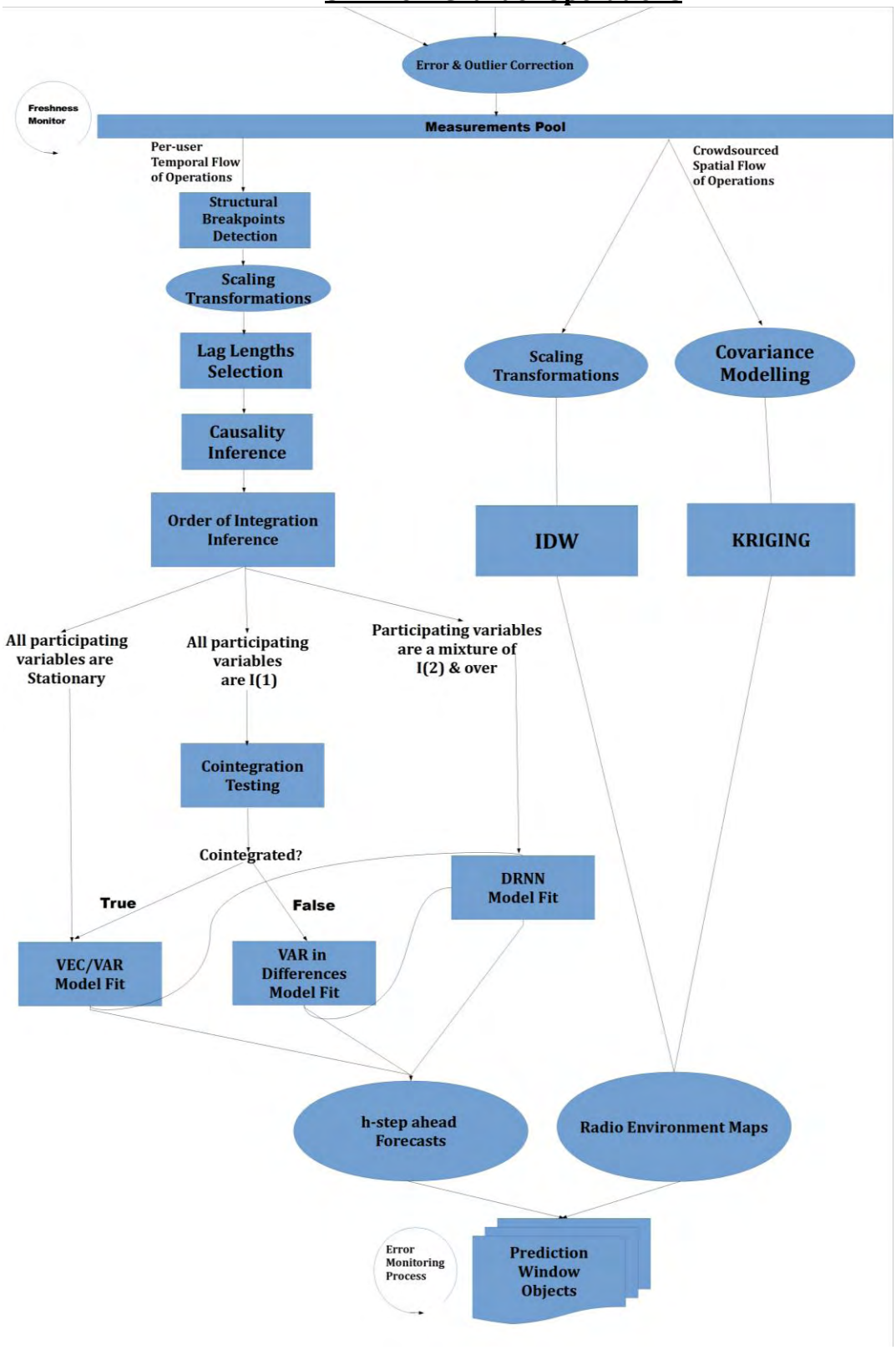

Figure 31: Flow Chart of Temporal and Spatial Operations

The flow of operations concerning the temporal and personalized study, shown on the left, is a prototype of a QoS monitoring and provisioning framework that is able to intelligently and swiftly perform a series of diagnostic tests and model selection procedures, to provide consistent model selection and estimation with the aim of offering time forecasts for the measures under study. Returning a timeline of forecasted 3GPP and non-3GPP QoS behaviour, the network can proactively schedule the Data Plane for each user, to effectively increase their combined bandwidth and to minimize each user's service access latency.

The flow of operations concerning the spatial study on the crowdsourced data, has been designed to generate 3GPP & non-3GPP REMs. The mentioned maps are available for utilization by applications that build around context-awareness and provisioning, such as Road Safety and Infotainment applications, while also assisting in the problem of MEC Service placement and migration, by offering the locations or migration destinations in which viable QoS can be achieved for a given vehicular user, known to demand ultra-reliable and continuous access to MEC Services, while being offered best latencies available. RAN-specific or region-specific Radio Environment Maps are thus available online while being frequently updated, on arrival of new appropriate measurements and departure of aged information.

The two flows are independent from each other i.e. can be executed separately, but for convenience in operation, they are fired at the same time i.e. completion of both constitutes a single iteration of our framework. In addition to REMs, diagnostic plots like dataset visualization, rolling statistics, ACF, NN training and validation loss per epoch, etc. are able to be generated, directly affecting our execution time, owing to their I/O intensive workload. Due to the computational complexity of the employed operations under the respective timing constraints and as permitted by the processing power of the hosting machine(s), the framework's code runs using phase-specific worker Processes, triggering shared Event objects on each phase's calculations completion, handing off their results to also shared structures between the main and secondary processes, subsequently exiting execution in a safe manner, to enable reusability of computation resources.

Chapter 6 – Experimentation

6.0: Setup Fundamentals

6.0.1: OpenAirInterface

Wireless Access Networks research can impose important prerequisites in terms of performance and flexibility. A network under implementation and testing has to be able to cope with ever-increasing mobile data traffic and low latencies, all while providing the freedom to augment and build onto with novel schemes. Controlled and scalable evaluations can be performed with **OpenAirInterface (OAI)** [44], an open experimentation and prototyping platform created by the Mobile Communications Dept. at EURECOM. It is comprised of software implementations of LTE and beyond systems, to be run on hardware-generic USRPs, designed according to 3GPP-imposed standardization efforts. The transceiver functionality of a component is realized via a software radio front end connected to a host computer for processing, following the **SDR (Software Defined Radio)** paradigm, from the UE point, up to the Core Network.

It maintains backward-compatibility with legacy 3GPP systems and is a useful tool for Network-driven research. It is essentially divided in two sub-repositories, one for the Core Network functionality and one for the Radio Access Network. It is written in C language and the compilation of the code is automated with the use of CMake Lists. Configuration scripts dictate the system parameters, such as allocated bandwidth, operating frequency, topology, IP-level information for interfaces, etc. This information is carefully parsed, transferred to runtime and from there on, one is free to implement and evaluate.

6.0.2: Porting R code to Python

Python's granted freedom for software development has evidently affected modern-era programming. Through higher-level abstraction, lacking little in content and options, it is fair to say that it is accessible by anyone who wishes to implement a solution for a given problem statement. While libraries for more recent technologies e.g. Tensorflow are widely available, there is still a gap to fill, concerning various mathematical methods and algorithms to match what the R community has to offer. Complete mathematical solutions and algorithms are conveniently packaged, easily installed and with manageable utilization. Connecting the above two, combines each language's virtues, enabling the construction of a Python-based framework that employs powerful R-powered solutions.

That is the point where *Rpy2* takes the spotlight. It comprises of a high-level interface to facilitate the use of R by Python programmers, exposing R objects as instances of Python-implemented classes, with R functions as bound methods to those objects for numerous cases. The numpy library can be seamlessly used for representing the data that are exchanged between the two via the interface. If one wishes to preserve the data

measurements in a dataframe structure, additional functionality is offered by the library to convert between a Python and an R dataframe representation.

One can block R code in the Python environment i.e. define an R function that is passed as a callable to the Python side, taking care of the input and output data conversion. This can be demonstrated with the following example of a structural breaks detection procedure:

```
import rpy2.robjects as robjects

bkps="""
    structbkps<-function(data,variable,lags,breaks){
        #Bai & Perron(2003) Structural Break Test for simultaneous estimation of
         #multiple unknown breaks
        a=as.data.frame(collapse::flag(x = data,n=0:lags,fill=c(0)),make.names=TRUE)
        names<-as.vector(colnames(a))
        names<-names[names!=variable]
        formulation<-paste(variable,paste(names,collapse=" + "),sep=" ~ ")
        form<-as.formula(formulation)
        bps<-strucchange::breakpoints(formula=form,data=a,breaks=breaks)
        title= paste(variable," Structural Breaks")
        if(is.null(bps$breakpoints)){ #If computed breaks do not minimize BIC
                                      #Criterion ,we safely return
           my_list=list(t = "zero","Formula"=formulation)
        }
        else{
            my_list=list(title = bps$breakpoints,"Formula"=formulation)
        }
        names(my_list)[1]<-title
        return (my_list)
    }
"""

BKPS=robjects.r(bkps)
```

A Python string of R code is assigned to a Python variable, which is in turn passed to the robjects.r module of rpy2, transcribing and converting our R function into a Python callable. Care must be taken to convert the Python Pandas dataframes into R native dataframes and vice-versa, through pandas2ri.py2rpy and pandas2ri.rpy2py modules, after activating the p2r module, as dictated by the library:

```
import rpy2.robjects.pandas2ri as p2r
p2r.activate()


r_list=BKPS(p2r.py2rpy(df),"mRssi",lags,numbreaks)
```

An important programming convention in this context is that if we have multiple return arguments from the R side, these must be formulated as R lists, in order to be decapsulated properly by the Python side, so as e.g. on return of the above Python-wrapped R function, we can retrieve its contents by numerical indexing. Furthermore, we can define a wrapper function for an implemented R procedure that is ported into Python, that assigns it to a multiprocessing.Process in order to conduct the computations in parallel. Our software was implemented in Python 3.6.3, R 3.6.3 and Rpy2 3.3.5.

6.0.3: NITOS Testbed

The NITOS Testbed, conceived and developed by NITlab (Network Implementation Testbed Laboratory) is comprised of 2 wireless testbeds for experimentation with heterogeneous technologies. The outdoor testbed features Wi-Fi, WiMAX and LTE support, located at the exterior of the University of Thessaly (UTH) campus building. The indoor isolated testbed is comprised of high-processing-powered and cutting-edge nodes, while being federated with the outdoor one. It is an important enabler of powerful experimentation and evaluation of implementations around algorithms and protocols in a large scale testbed, all with high availability and reproducibility of results.

Figure 32: Outline of the NITOS infrastructure

6.1: Experimentation and Evaluation

6.1.1: Proposed Experimental Setup

Our experimentation setup is based on the one presented in [18], where multiple NITOS testbed nodes are employed for accommodating our 5G MEC-Enabled Heterogeneous Network components, starting from the Core Network and ending at the multi-homed UEs. In this context, the high-processing powered MEC-Enabled Host conveniently homes all MEC Agent, MEC Services and AI operations, while the latter can be separately executed since relying only on the fetched measurements from a remote database, residing in close proximity to our nodes. The wireless configuration is comprised of 40 MHz IEEE 802.11n channels in 2.4 GHz and 5MHz SISO mode for the 3GPP case.

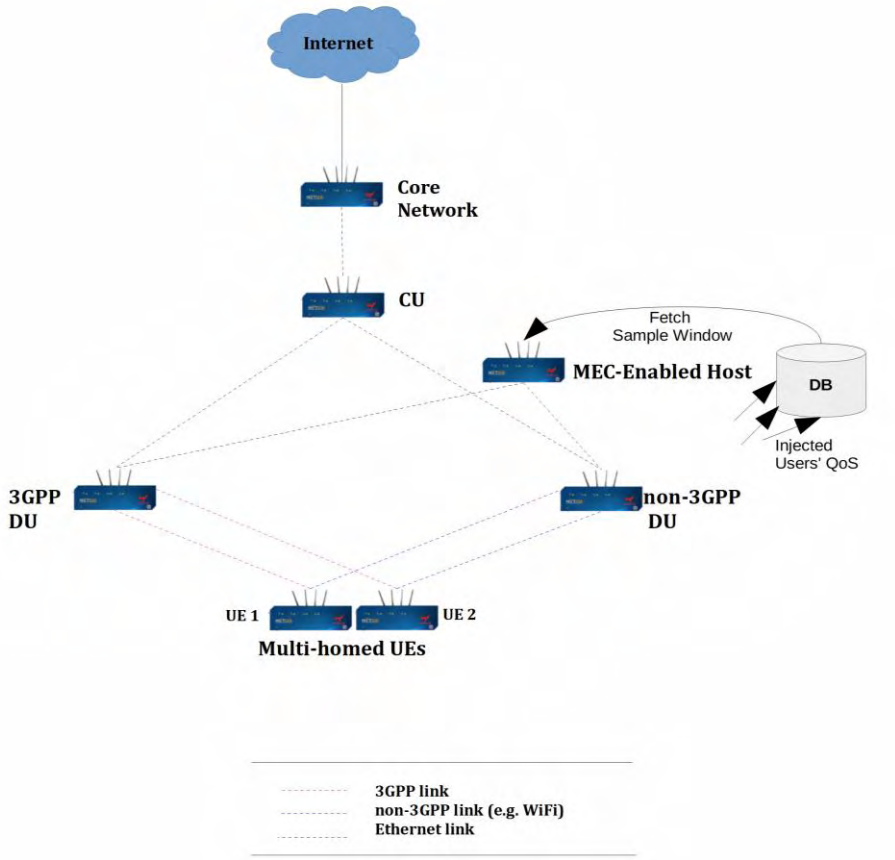

Figure 33: Proposed Experimental Setup

6.1.2: Software Performance

Calculations for the results shown below were all carried out on nodes that employ 8-core Intel CPUs, operating at 3.4 GHz and without GPU support, yielding a swift operations cycle time of ~1min for both spatial and temporal flows of operations, as depicted by Figure 31.

6.1.3: Implemented approaches Demonstration

Our example dataset includes various moderate-speed trajectories of dually-connected and identical in device specifications vehicular users, reporting their QoS at a frequent rate. The results showcased here are based on data provided from the respective drive tests, since our testbed experimentation setup currently allows only for the stationary mobility case. We utilize multiple users' behaviour for the study in the spatial domain, creating 3GPP REMs for RSSI, RSRQ, RSRP and RSSNR metrics and non-3GPP WiFi-specific RSSI REMs, with a resolution of 35000 points. We will also study user-specific behaviour to showcase the flexibility of our framework in modelling and predicting.

6.1.3.1: Dual-Technology REM examples

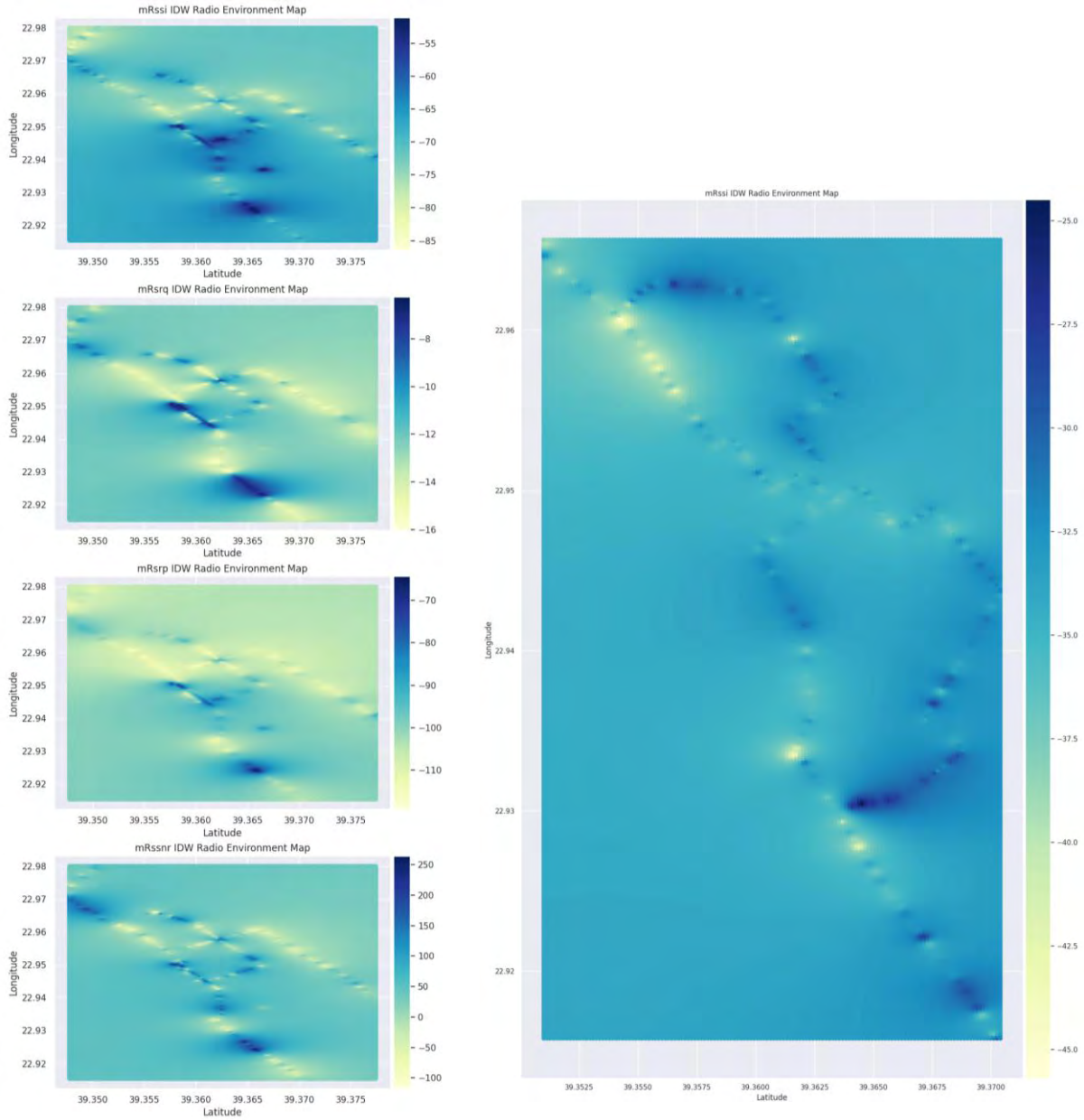

Figure 34: 3GPP & non-3GPP IDW Crowdsourced REMs

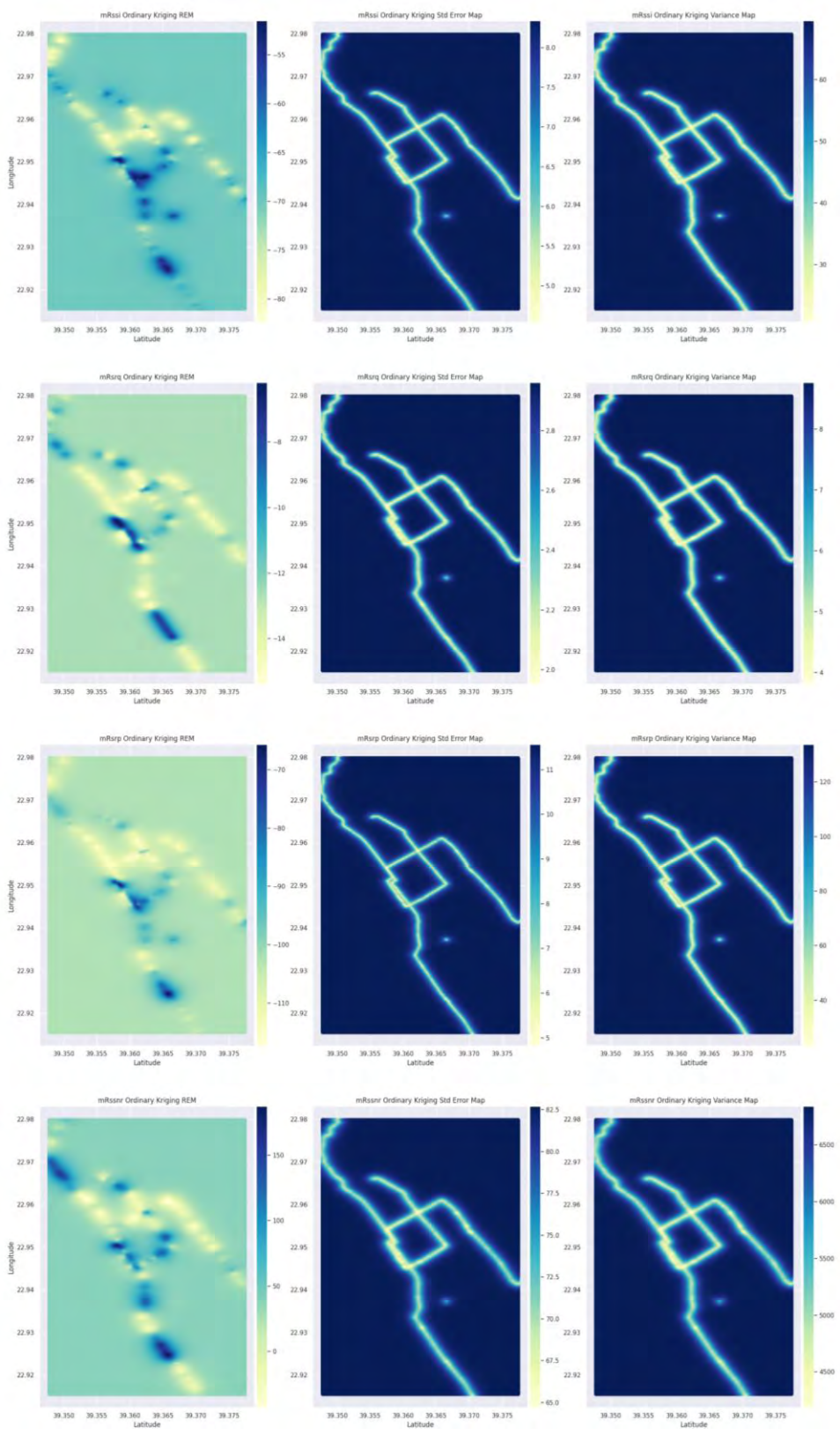

Figure 35: 3GPP Ordinary Kriging Crowdsourced REMs

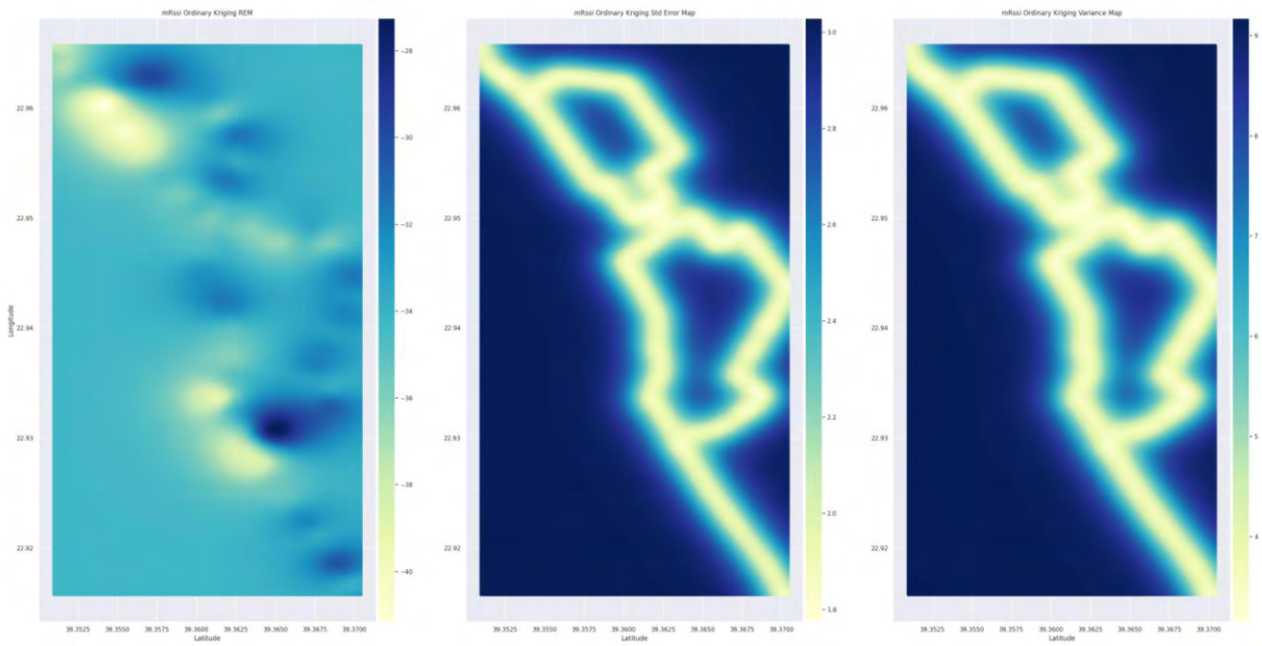

Figure 36: non-3GPP Ordinary Kriging Crowdsourced REMs

We can visually confirm that allowing multiple users to enter our grid of measurements can lead to more information for the covariance modelling that is performed for the Kriging case, while also making IDW REMs more detailed. One can also observe that in the IDW case, the chosen power factor of 2 appears to let more spatial points to enter each neighbourhood, pointing to the need for a fine-tuning procedure for the power factor. The resulted errors for the 3GPP Ordinary Kriging measurements show that RSSI, RSRQ and RSRP can be efficiently predicted over a grid map in the spatial domain, all while maintaining small errors. In contrast, the RSSNR metric seems to suffer from large errors and seems unsuitable to be taken into account by our case. The resulted errors for the non-3GPP case also appear well-behaved.

6.1.3.2: User-specific 3GPP QoS Time Forecasting example

For the temporal case, we will focus on the findings for the sample data of a single vehicular user that frequently reports his/her 3GPP QoS, as plotted in earlier sections. Our aim is to assess how a correctly specified VAR model can perform in terms of accompanying error, concerning multi-step forecasts of future user-specific QoS readings, as compared to models of higher complexity. In this way we can examine whether long-term forecasting is viable in our context and use case.

Having meticulously prior described our Temporal Flow of Operations, we will focus on the findings alone for a single-user sample, as plotted in earlier sections. After deducting that our sample is structural breakpoint(s)-free, our framework suggests a lag length of p=4, obtained after testing for serial correlation in the residuals with the Breusch-Godfrey approach, as per our Lag Lengths Selection Phase. I(d) inferencing Phase suggests I(1) for all 4 3GPP metrics and most causalities are bidirectional. Indicating possible cointegration, the Johansen approach finally gives light to our study, suggesting full rank for all 0.90,0.95,0.99 cut-off values. Formulating a stable & Cointegrated VAR(4) for our metrics, we perform multiple time forecasts of varying forecast horizons and compare their adequacy in the Relative Error sense, with another approach. The latter essentially is a VAR-DRNN formulation, where the Neural Network can utilize an augmented dataset, as offered by a VAR model, enabling forecasting for multiple steps ahead in time using a multivariate approach, typically expected to leverage the NN's forecasting power, compared to the univariate case. Further, we note that the formulation of multiple DRNNs is performed in a fully automated manner, essentially having one DRNN for each metric of interest by creating a supervised input structure according to the findings of the previous analysis that was performed under the VAR framework i.e. including 4 lags of the variables that seem to be providing statistically significant information in forecasting our 3GPP metric of interest. The neural networks are setup using a MAE loss function, with NADAM optimization and 4 lightweight Layers of RNN specifications i.e. one LSTM and 3 GRU Layers, employing IHS activation functions. Prior to formulating, our variables are scaled analogously, after being subtracted their mean vector. We estimate forecasts up to 45 points ahead in time, back-transform and assess their error evolution across expanding forecast horizons.

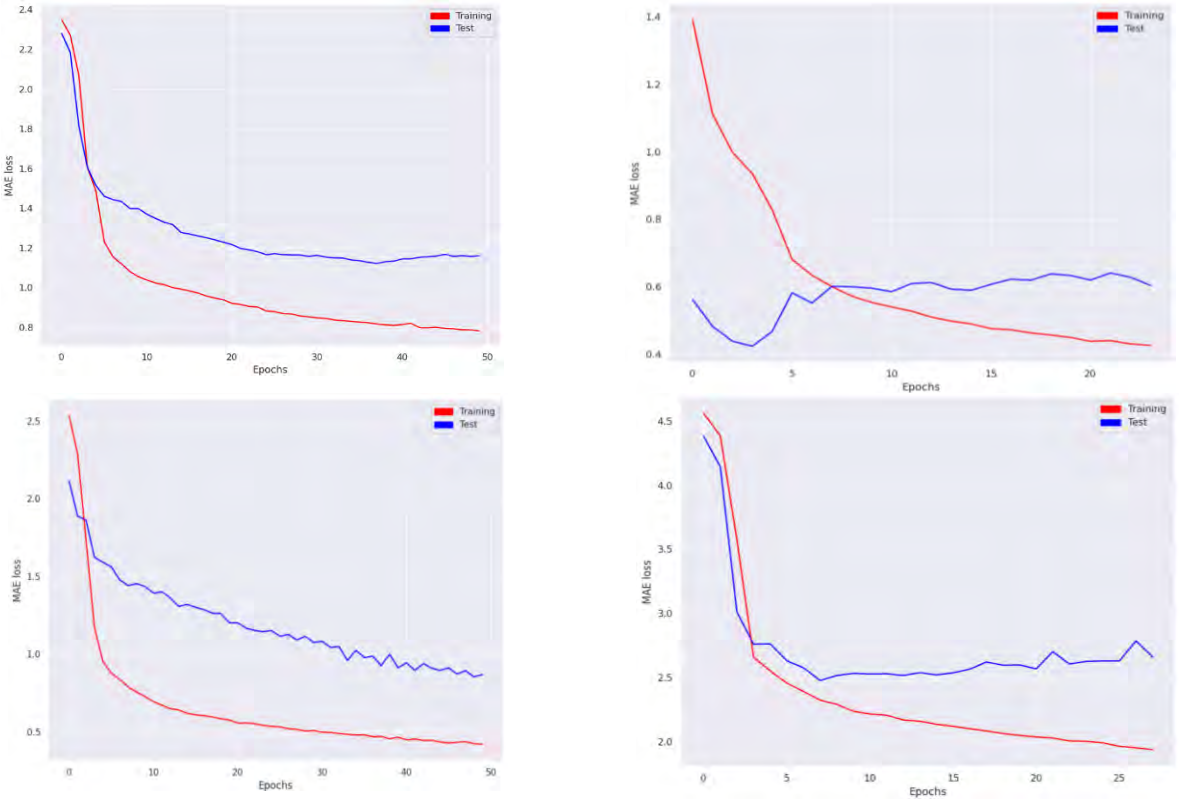

Figure 37: Example MAE Loss per epoch plots for lightweight DRNN fits

Depicted above, are example loss per epoch plots for lightweight DRNN fits of our configuration. Metrics mRssi (top left), mRsrq (top right), mRsrp (bottom left) and mRssnr (bottom right), seem well-behaved in terms of their loss evolution across the batch training epochs, that are upper-bounded by a fixed value, due to the desired timing constraints imposed e.g. in this configuration max epochs is set to 50. Any observed overlaps of the loss curves for the initial epochs can owe to the nature of the validation data, as compared to the training data. Indeed, the loss curves seem to nevertheless continue in a progressively stabilizing manner, when we note that each deployed DRNN stops at its own pace and the way it stops is directly controlled by the *patience* option of the Early Stopping feature. To elaborate, mRssi and mRsrp utilize all of their permitted epochs, while mRsrq and mRssnr stop prematurely due to the patience value imposed. In such case, we may allow for a higher patience value, to potentially benefit the latter two metrics. We should also note that the training/validation partitioning of a sample, the loss function employed (other options may include LogCosh), as well as the utilized optimizer can directly impact the loss curves for the training and validation sets. We proceed with comparing our resulted stable and Cointegrated VAR(4) model with the VAR-DRNN formulation described, across increasing forecast horizons, to address their suitability for lengthy forecasts, as needed by our problem statement. As focusing on the actual vs forecasted plots is highly dependent on the utilized sample and the dataset partitioning, often leading to dataset-specific, result-driven trial and error approaches, we believe that a sensible approach to the matter of quantifying our predictive power is examining how the forecasting errors evolve for increasing forecast horizons as below:

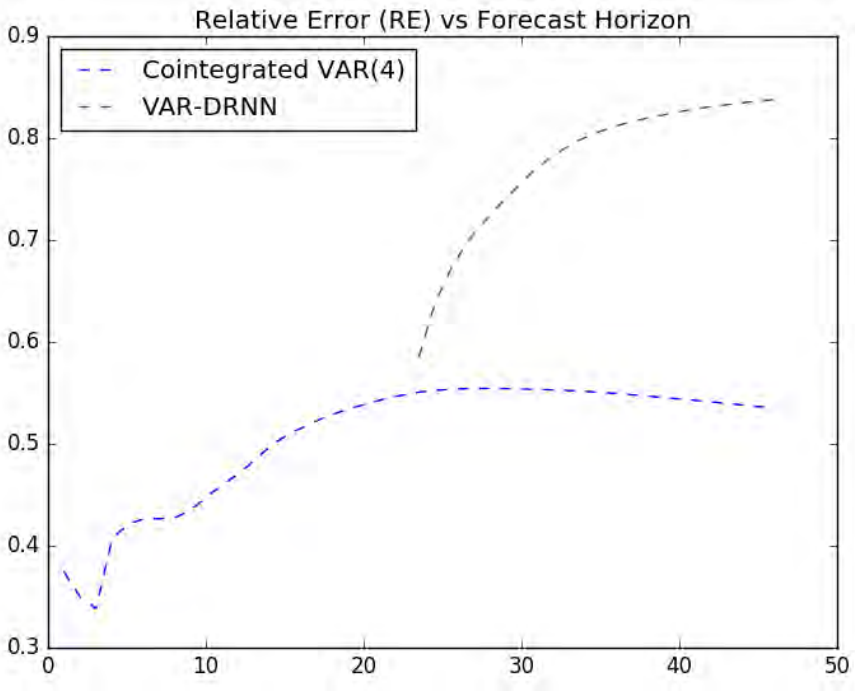

Figure 38: Comparison in terms of RE for varying forecast horizons

The errors reported in the graph above, are calculated using the back-transformed original data scales. We can visually confirm that correct model specification and proper testing, lead to seemingly simpler models that can efficiently tackle the problem of multivariate multi-step time forecasting. The first 22 points of the VAR-DRNN line were omitted from the plot, since they contributed to the weighting of the Neural Networks, being the validation set. More complex modelling can thus be prone to failing in further leveraging our forecasting power and careful fine-tuning techniques need to be carried out e.g. modifying the optimizer's learning rate on certain validation loss behaviours. The particular VAR-DRNN approach is thus completely outperformed by a theory-backed stable and Cointegrated VAR model, which, in turn, seems to be applicable to our problem of long-term forecasting of future user-specific QoS measurements. Nonetheless, this example should serve as indicative of our framework's capabilities in automated and swift formulation of multiple models with the aim of forecasting i.e. there is more than enough room for improvement on the error behaviours of the VAR-DRNN approach, while further examination on how the errors relate to each decision made in our framework's configuration is something that requires case-specific testing which can be carried out as future work.

6.2: Conclusion and Future Work

In this thesis, we first provided an outline of the Radio Access Network evolution across the cellular generations. With a tendency to be continuously disaggregated and virtualized, effectively leveraging our network's flexibility and scalability, 5G networks are faced with the challenge of providing a fertile ground for pioneering next-generation verticals, as in the case of V2X communications. That essentially means, as per our problem statements, that RAT switching must be performed proactively, so as users can be offered the best radio conditions available and also any services they access must be effectively live migrated, according to the user trajectories, so as user session states are transferred, while granting no service disruptions and zero downtime.

Through an exploratory of mathematical and time series-specific terminology, VAR models and Neural Networks, we were able to infer, model and forecast a user's QoS in time through a powerful multi-processing framework that is for one, able to meticulously test and remedy for various time series pathogenies. Performing the temporal flow of operations for both technologies, can offer a reasonable "peek" into a moderate-sized future window, enabling us to proactively schedule a specific user's data plane by continuously offering what is deduced as best, in terms of channel quality and service access latency, while maintaining non-prohibiting errors. In addition, this framework employs covariance modelling and spatial interpolation procedures, to create up-to-date Radio Environment Maps, available to applications that can utilize this information for reasons of transport safety and performance monitoring and provisioning, such as Road Safety and/or Infotainment applications. The resulted maps for both 3GPP and non-3GPP technologies, can provide an up-to-date collaborative view of the QoS conditions in our RAN, effectively pinpointing areas where users can experience viable and adequate QoS, with respect to the services that the users access. On need of a service migration decision, these maps can be queried, either on whole or for specific areas, to quantify on the suitability of a possible migration destination.

Additional future goals to the ones mentioned throughout this thesis' sections, would include incorporating more approaches in our framework for the temporal/spatial operations and also employing more sophisticated ML formulations such as spatiotemporal models. Further intelligence for cutting back on computational costs is another subject that requires addressing, always with the aim of continuously learning, testing and further progressing, in an attempt to meet and expand our current boundaries and learn from what can work best in a new real-world setting.

References

[1] - Marsch et al. “5G System Design”, Wiley
[2] - umtsworld.com/technology/overview.htm
[3] - researchgate.net/publication/265014886_4G_Wireless_Technology_A_Brief_Review
[4] - tutorialspoint.com/lte/lte_protocol_stack_layers.htm
[5] - lteexpert.blogspot.com/2014/10/mac-pdu-formats.html
[6] - artizanetworks.com/resources/tutorials/lay_2_str.html
[7] - artizanetworks.com/resources/tutorials/lay_2.html
[8] - artizanetworks.com/resources/tutorials/lay_2_log.html
[9] - onlinelibrary.wiley.com/doi/abs/10.1002/9780470742891.ch16
[10] - 3gpp.org/technologies/keywords-acronyms/98-lte
[11] - N. Makris, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas: “Experimental Evaluation of Functional Splits for 5G Cloud-RANs”
[12] - N. Makris, C. Zarafetas, P. Basaras, T. Korakis, N. Nikaein and L. Tassiulas: "Cloud-Based Convergence of Heterogeneous RANs in 5G Disaggregated Architectures"
[13] - A. Machen, S. Wang, K. K. Leung, B. J. Ko and T. Salonidis: “Live Service Migration in Mobile Edge Clouds”
[14] - C. Chang, K. Alexandris, N. Nikaein, K. Katsalis, T. Spyropoulos: “MEC architectural implications for LTE/LTE-A networks”
[15] - N. Makris, V. Passas, T. Korakis, L. Tassiulas: “Employing MEC in the Cloud-RAN: An Experimental Analysis”
[16] - K. Gillani, J. Lee: “Comparison of Linux virtual machines and containers for a service migration in 5G multi-access edge computing”
[17] - D. Harutyunyan, R. Riggio: “Flexible functional split in 5G networks”
[18] - N. Makris, V. Passas, C. Nanis and T. Korakis: "On Minimizing Service Access Latency: Employing MEC on the Fronthaul of Heterogeneous 5G Architectures"
[19] - F. Giust, M. Filippou: “Multi-access Edge Computing: The driver behind the wheel of 5G-connected cars”
[20] - M. Emara, M. Filippou, D. Sabella: “MEC-aware Cell Association for 5G Heterogeneous Networks”
[21] - K. Zhang et al.: “Energy-efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks”
[22] - S. Wang et al.: “Survey on Service Migration in MEC”
[23] - C. Puliafito et al.: “Virtualization and Migration at the Network Edge: an Overview”
[24] - V. Medina, J.M. Garcia: “Survey on Migration Mechanisms of Virtual Machines”
[25] - arimas.com/78-rsrp-and-rsrq-measurement-in-lte
[26] - A. B. Shams et al.: “Mobility Effect on the Downlink Performance of Spatial Multiplexing Techniques under Different Scheduling Algorithms in Heterogeneous Network”
[27] - etsi.org/images/files/ETSIWhitePapers/etsi_wp23_MEC_and_CRAN_ed1_FINAL.pdf
[28] - S. Ali et al.: “6G White Paper on Machine Learning in Wireless Communication Networks”
[29] - A. Robins: “Catastrophic forgetting, rehearsal, and pseudorehearsal”, Journal of Neural Computing, Artificial Intelligence and Cognitive Research, 1995

[30] - Y.P. Hsu, E. Modiano, L. Duan: "Age of Information: Design and Analysis of Optimal Scheduling Algorithms"
[31] - J. Riihijärvi, P. Mähönen: "Machine Learning for Performance Prediction in Mobile Cellular Networks"
[32] - umass.edu/landeco/teaching/multivariate/readings/McCune.and.Grace.2002.chapter9.pdf
[33] - N. Perpinias, A. Palaios, J. Riihijärvi, P. Mähönen: "A Measurement-Based Study on the Use of Spatial Interpolation for Propagation Estimation"
[34] - J. Liao et al: "Radio Environment Map Construction by Kriging Algorithm Based on Mobile Crowd Sensing"
[35] - S. Sinharay: "International Encyclopedia of Education (Third Edition), 2010"
[36] - itl.nist.gov/div898/handbook/eda/section3/eda35c.htm
[37] - web.vu.lt/mif/a.buteikis/wp-content/uploads/2020/02/Lecture_01.pdf
[38] - H. Y. Toda, T. Yamamoto: "Statistical inference in vector autoregressions with possibly integrated processes"
[39] - C. W. J Granger, P. Newbold: "Spurious regressions in econometrics"
[40] - karlwhelan.com/Teaching/MA%20Econometrics/part4.pdf
[41] - D. Giles: "Spurious Results with Time-Series Data: Further Asymptotic Results"
[42] - etsi.org/deliver/etsi_gs/MEC/001_099/012/02.01.01_60/gs_MEC012v020101p.pdf
[43] - D. Kwiatkowski et al: "Testing the null hypothesis of stationarity against the alternative of a unit root. How sure are we that economic time series have a unit root?"
[44] - N. Nikaein et al: "OpenAirInterface: A Flexible Platform for 5G Research"
[45] - wiki.pathmind.com/neural-network
[46] - R. Pascanu et.al: "On the difficulty of training Recurrent Neural Networks"
[47] - Y. Bengio et.al: "Learning long-term dependencies with gradient descent is difficult"
[48] - J. Chung et al.: "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling"
[49] - R. Fu et al: "Using LSTM and GRU neural network methods for traffic flow prediction"
[50] - K. Cho et al: "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation"
[51] - S. Hochreiter et al: "Long Short-Term Memory"
[52] - M. H. Pesaran et al: "An Autoregressive Distributed Lag Modelling Approach to Cointegration Analysis."
[53] - Z. He, K. Maekawa: "On spurious Granger causality"
[54] - O. Simpson et al : "LTE RSRP, RSRQ, RSSNR and local topography profile data for RF propagation planning and network optimization in an urban propagation environment"
[55] - cran.r-project.org/web/packages/ARDL/ARDL.pdf
[56] - H. Goel, I. Melnyk, A. Banerjee: "R2N2: Residual Recurrent Neural Networks for Multivariate Time Series Forecasting"
[57] - S. Ng, P. Perron: "Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power"
[58] - lancaster.ac.uk/pg/waller/pdfs/Intermittent_Demand_Forecasting.pdf

Term	Meaning
5G	5th Generation
RAN	Radio Access Network
RAT	Radio Access Technology
VR	Virtual Reality
AR	Augmented Reality
V2X	Vehicle-to-Everything
URLLC	Ultra-Reliable Low-Latency Communications
3GPP	3rd Generation Partnership Project
LTE	Long-Term Evolution
MEC	Multiple-Access Edge Computing
NITOS	Network Implementation using Open-Source software
USRP	Universal Software-Defined Radio Peripherals
QoS	Quality of Service
AI	Artificial Intelligence
REM	Radio Environment Map
NITLAB	Network Implementation Testbed LABoratory
KPI	Key Performance Indicator
PDCP	Packet Data Convergence Protocol
RLC	Radio Link Control
F1AP	F1 Application Protocol
VM	Virtual Machine
AR(p)	AutoRegressive process of order P
ADF	Augmented Dickey-Fuller
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
VAR(p)	Vector AutoRegressive Process of order P
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
IDW	Inverse Distance Weighting
GSM	Global System for Mobile communications
UMTS	Universal Mobile Telecommunications System
E-UTRAN	Evolved Universal Mobile Telecommunications System Terrestrial Radio Access
DU	Distributed Unit
eNB	evolved Node B
ACF	AutoCorellation Function
PACF	Partial AutoCorellation Function
IHS	Inverse Hyperbolic Sine
ICT	Information and Communications Technology
eMBB	enhanced Mobile BroadBand

mMTC	massive Machine-Type Communications
eLTE	enhanced Long-Term Evolution
Wi-Fi	Wireless-Fidelity
UX	User eXperience
IMT	International Mobile Telecommunications
DL	DownLink
UL	UpLink
GHz	GigaHertz
CDF	Cumulative Density Function
RTT	Round Trip Time
MAC	Medium Access Control
MHz	MegaHertz
Gbps	Gigabits per second
CS	Circuit Switching
PS	Packet Switching
MS	Mobile Base Station
BSS	Base Station Subsystem
NSS	Network & Switching Subsytem
OSS	Operation and Support Subsystem
BTS	Base Transceiver Station
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
MSC	Mobile Services switching Center
PSTN	Public Switched Telephone Network
ISDN	Integrated Services Digital Network
GPRS	General Packet Radio Services
WCDMA	Wideband Code Division Multiple Access
FDD	Frequency Division Duplex
TDD	Time Division Duplex
CN	Core Network
UTRAN	UMTS Terrestrial Radio Access Network
UE	User Equipment
RNC	Radio Network Controller
RRC	Radio Resource Control
PLMN	Public Land Mobile Network
VLR	Visitor Location Registry
SGSN	Serving GPRS Support Node
GGSN	Gateway GPRS Support Node
EIR	Equipment Identity Registry
HLR	Home Location Registry
ATM	Asunchronous Transfer Mode

NPDB	Number Portability DataBase
IMSI	International Mobile Subscriber Identity
TMSI	Temporary Mobile Subscriber Identity
P-TMSI	Packet Temporary Mobile Subscriber Identity
TLLI	Temporary Logical Link Identity
MS-ISDN	Mobile Station ISDN
IMEI	International Mobile Station Equipment Identity
IMEI-SV	International Mobile Station Equipment Identity and Software Version
USIM	User Identity Module
SAE	System Architecture Evolution
HSPA	High-Speed Packet Access
HSDPA	High-Speed Downlink Packet Access
HSUPA	High-Speed Uplink Packet Access
MIMO	Multiple Input Multiple Output
IP	Internet Protocol
WLAN	Wireless Local Area Network
OFDM	Orthogonal Frequency Division Multiplexing
SC-FDMA	Single-Carrier Frequency Division Multiplexing Access
PAPR	Peak-to-Average Power Ratio
SNR	Signal to Noise Ratio
SGW	Serving GateWay
MME	Mobile Management Entity
PDN P-GW	Packet Data Network Gateway
HSS	Home Subscriber Server
PCRF	Policy Control and Charging Rules Function
EPC	Evolved Packet Core
SGW	Serving GateWay
NAS	Non-Access Stratum
PDN	Packet Data Network
DHCP	Dynamic Host Configuration Protocol
PCEF	Policy Control Enforcement Function
SDU	Service Data Unit
PDU	Protocol Data Unit
SN	Sequence Number
RB	Radio Bearer
SRB	Signal Radio Bearer
DRB	Data Radio Bearer
DCCH	Dedicated Control Channel
DTCH	Dedicated Traffic Channel
ARQ	Automatic Repeat reQuest

UM	Unacknowledged Mode
TB	Transport Block
HARQ	Hybrid Automatic Repeat reQuest
PHY	Physical
RF	Radio Frequency
BCCH	Broadcast Control Channel
PCCH	Paging Control Channel
CCCH	Common Control Channel
MTCH	Multicast Traffic Channel
BCH	Broadcast Channel
DL-SCH	Downlink Shared Channel
DRX	Discontinuous Reception
PCH	Paging Channel
MCH	Multicast Channel
MBSFN	Multicast-Broadcast Single-Frequency Network
MBMS	Multimedia Broadcast Multicast Service
UL-SCH	Uplink Shared Channel
RACH	Random Access Channel
PDSCH	Physical Downlink Shared Channel
PDCCH	Physical Downlink Control Channel
PHICH	Physical HARQ Indicator Channel
ACK	Acknowledgement
NACK	Non-Acknowledgement
PCFICH	Physical Control Format Indicator Channel
PBCH	Physical Broadcast Channel
P-SS	Primary Synchronization Signal
S-SS	Secondary Synchronization Signal
PUSCH	Physical Uplink Shared Channel
CQI	Channel Quality Indicator
PUCCH	Physical Uplink Control Channel
PRACH	Physical Random Access Channel
DM RS	Demodulation Signal
SRS	Sounding Reference Signal
TTI	Transmission Time Interval
NR	New Radio
CU	Central Unit
GTP	GPRS Tunneling Protocol
SCTP	Stream Control Transmission Protocol
F1oIP	F1 over IP
SDN	Software Defined Networking
NFV	Network Functions Virtualization

SLA	Service-Level Agreement
KVM	Kernel-based Virtual Machine
OS	Operating System
ETSI	European Telecommunications Standards Institute
RNTI	Radio Network Temporary Identifier
U-DS	User-Dedicated Service
RNIS	Radio Network Information Service
V2I	Vehicle-to-Infrastructure
LAN	Local Area Network
ARP	Address Resolution Protocol
WAN	Wide Area Network
VPN	Virtual Private Network
DNS	Domain Name System
SAN	Storage Area Networks
NAS	Network Attached Storage
NFS	Network File System
CPU	Central Processing Unit
SSH	Secure Shell
DGP	Data Generating Process
RSU	RoadSide Unit
AoI	Age of Information
RSRP	Reference Signal Received Power
RS	Reference Signal
RE	Resource Entity
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RSSI	Received Signal Strength Indicator
RSRQ	Reference Signal Received Quality
RSSNR	Received Signal to Noise Ratio
SINR	Signal-to-Interference Noise Ratio
macroBS	macro Base Station
MTS	Multiple Time Series
UTC	Coordinated Universal Time
OLS	Ordinary Least Squares
MICE	Multivariate Imputation via Chained Equations
IC	Information Criterion
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
RSS	Residual Sum of Squares
CLT	Central Limit Theorem
MA(p)	Moving Average process of order P

ARIMA	AutoRegressive Integrated Moving Average
WN	White Noise
SC	Schwartz Criterion
HQ	Hannan-Quinn Criterion
MLE	Maximum Likelihood Estimation
DF	Dickey-Fuller
PP	Phillips-Perron
ZA	Zivot-Andrews
ARDL	AutoRegressive Distributed Lags
TYGC	Toda-Yamamoto Granger Causality
I(d)	Integration Order of d
H0	Null Hypothesis
H1	Alternative Hypothesis
TS	Trend Stationarity
DS	Difference Stationarity
ECM	Error-Correction Model
VEC	Vector Error Correction
DNN	Deep Neural Network
NN	Neural Network
NLP	Natural Language Processing
ANN	Average Nearest Neighbours
RE	Relative Error
OAI	OpenAirInterface
UTH	University of Thessaly