Politechnika Wrocławska

FIELD OF SCIENCE: Engineering and Technology

DISCIPLINE OF SCIENCE: Information and Communication Technology

DOCTORAL DISSERTATION

AI-assisted dimensioning methods for Network Slicing in Next Generation Mobile Networks

Dominik Dulas

Supervisor:

Prof. dr hab. inż. Krzysztof Walkowiak

October 2024

"The size of your dreams must always exceed your current capacity to achieve them. If your dreams do not scare you, they are not big enough."

This Child Will Be Great: Memoir of a Remarkable Life by Africa's First Woman President
by Ellen Johnson Sirleaf

Acknowledgements

I would like to start by extending my heartfelt appreciation to my supervisor, Prof. Krzysztof Walkowiak, for his guidance and unwavering support throughout my Ph.D. studies over the past four years. His vast knowledge, which he generously shared, was instrumental in the completion of this thesis and other publications. I am grateful for his trust, patience, motivation, understanding, and insightful feedback.

I am truly grateful to Prof. Agnieszka Wyłomańska, PhD candidates Justyna Witulska and Katarzyna Maraj-Zygmąt from the University of Science and Technology in Wroclaw, Poland, and to Prof. Ireneusz Jabłoński from the Brandenburg University of Technology and Fraunhofer Institute for Photonic Microsystems, Cottbus, Germany for the opportunity to collaborate on our research and co-author multiple papers together.

I would like to express my genuine gratitude to the Network and Performance Engineering department at Nokia, with particular recognition to the Leadership Team headed by Sebastian Lasek and my company patron for the Industrial PhD, Marcin Grygiel, for fostering an excellent and cooperative work environment.

I would like to sincerely thank my Telco Data Science team, technically lead by PhD candidates Michał Panek and Jakub Mazguła for a wonderful team spirit, great competences and outstanding collaboration.

Lastly, I extend my heartfelt gratitude to my wife Małgosia and daughter Basia for the love, support, and joy you continuously provide.

The research presented in this dissertation has been supported by funding from the Polish Ministry of Education and Science via the Industrial PhD program "Doktorat wdrożeniowy II - sztuczna inteligencja" as a part of the project: DWD/4/2/2020.

Abbreviations

ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AR	Augmented Reality
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
BH	Busy Hour
BIC	Bayesian Information Criterion
BiLSTM	Bidirectional Long Short-Term Memory
BLER	Block Error Rate
BTS	Base Transceiver Station
CapEx	Capital Expenditures
CDF	Cumulative Distribution Function
Cloud-RAN	Cloud Radio Access Network
CNN	Convolutional Neural Network
CNN-BiLSTM	Convolutional Neural Network-Bidirectional Long Short-Term Memory
CPU	Central Processing Unit
CQI	Channel Quality Indicator
CSP	Communication Service Provider
CSPs	Communication Service Providers
CU	Centralized Unit
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
DT	Digital Twin
DTW	Dynamic Time Warping
DU	Distributed Unit
DV	Data Volume
eMBB	Enhanced Mobile Broadband

FTP	File Transfer Protocol
FTSM	Foundation Time Series Model
GP	Gaussian Process
HQIC	Hannan-Quinn Information Criterion
HW	Hardware
IAB	Integrated Access and Backhaul
KPI	Key Performance Indicator
LLM	Large Language Model
LSTM	Long Short-Term Memory
LTE	Long-Term Evolution
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MG	Multiplexing Gain
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLE	Maximum Likelihood Estimation
mMTC	massive Machine Type Communications
MNOs	Mobile Network Operators
MSE	Mean Squared Error
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Infrastructure
NGMN	Next Generation Mobile Networks
nMAE	normalized Mean Absolute Error
NPN	Non-Public Network
NS	Network Slicing
PCA	Principal Component Analysis
PNF	Physical Network Functions
PRB	Physical Resource Block
QoS	Quality of Service
RAN	Radio Access Network
RMSE	Root Mean Square Error

RNN	Recurrent Neural Network
RU	Radio Unit
SARIMA	Seasonal AutoRegressive Integrated Moving Average
SLA	Service Level Agreements
SVM	Support Vector Machines
SW	Software
TES	Thresholded Exponential Smoothing
TM	Traffic Model
TSC	Time-Sensitive Communication
TTI	Transmission Time Interval
UEs	User Equipments
URLLC	Ultra-Reliable, Low-Latency Communications
VARMA	Vector Autoregressive Moving-Average
VARMAX	Vector Autoregressive Moving Average with eXogenous regressors model
VNF	Virtual Network Functions
VR	Virtual Reality

Abstract

This doctoral dissertation explores the dimensioning of 5G mobile network technology, with a specific focus on incorporating the impacts of network slicing. The research carried out in the context of this dissertation contributes to the advancement of techniques for dimensioning 5G networks. It is believed that as the complexity of the 5G architecture rises, the manual tasks performed by technical experts will no longer suffice for input preparation, such as traffic modeling, making it essential to develop AI-based methods.

As a result of the research, a new methodology and framework was developed that considers the:

1. key performance indicators selection,
2. performance forecasting,
3. predictive modeling for regression of seledcted outputs (e.g. throughput and delay),
4. indirect estimation of link capacity,

which will be used in Nokia's network planning and dimensioning processes. The use of real network data to develop and verify the models and algorithms created adds to this innovation.

Forecasting throughput and delay is an important component of the framework that allows indirect dimensioning of 5G BTS capacity. As part of the research, the use of multivariate predictive models was performed to forecast slice level throughput and delay as a data-driven approach to dimension 5G capacity. After comprehensive comparison, the VARMAX model, a vector autoregressive moving average model with additional exogenous inputs, was selected as the best model to forecast throughput and delay. The results indicate that this model is equally effective for short- and long-term predictions with commendable accuracy. Additionally, incorporating configurational knowledge, such as the frequency band, into the model's training process enhances its accuracy. The evaluation of one-dimensional models for the forecasting of environmental variables was also performed as a supporting element for the multivariate model. For this problem a Lag-Llama model, which is a foundational time series model, was selected after a thorough evaluation. All validations and comparisons were made with normalized mean absolute error and mean absolute percentage error metrics.

In addition, this work presents an original technique, using system-level traffic data, to estimate the statistical multiplexing gain of aggregated 5G transport links. The algorithm enables the scalability of the simulation outcomes. This approach reduces the computational time from days to seconds, which is crucial for network planning recommendations, and ultimately improves the efficiency and flexibility of services provided to telecommunication operators. Two case studies have been presented, demonstrating the alignment of the estimations with measured values from microwave links in mobile networks and highlighting their relevance to cloud BTS dimensioning.

Finally, a data-centric framework is introduced for forecasting and dimensioning, integrating the digital twin concept. This model can autonomously serve as a forecasting tool for (sliced) network dimensioning and traffic management, or it can act as a key component of a comprehensive digital twin. In addition, it illustrates the feasibility of how interconnected methods investigated in this work deliver the necessary output. To verify the validity of the framework and evaluate its applicability and ability to maintain the physical context, experiments were performed on the actual data. The results show that the proposed framework can effectively elucidate and quantify these phenomena through data-driven simulations of sliced wireless networks. Implementing the framework will reliably assist Nokia processes by automatically recommending capacity expansions or configuring parameters for slice planning based on the real data.

Streszczenie

Niniejsza rozprawa doktorska bada wymiarowanie sieci komórkowej 5G, ze szczególnym uwzględnieniem wpływu plastrowania sieci (ang. Network Slicing). Raportowane analizy oraz ich wyniki przyczyniają się do rozwoju technik wymiarowania sieci 5G. Uważa się, że wraz ze wzrostem złożoności architektury 5G, ręczne zadania wykonywane przez ekspertów technicznych nie będą już wystarczające do przygotowania danych wejściowych, takich jak modelowanie ruchu, co czyni koniecznym opracowanie metod opartych na sztucznej inteligencji.

W wyniku badań opracowano nowe, kompleksowe rozwiązanie (ang. framework), które uwzględnia:

1. wybór kluczowych wskaźników wydajności,
2. prognozowanie wydajności,
3. modelowanie predykcyjne do regresji wybranych wyników (np. przepustowości i opóźnienia),
4. pośrednie oszacowanie przepustowości łącza,

które będzie wykorzystywane w procesach planowania i wymiarowania sieci Nokii. Zastosowanie rzeczywistych danych sieciowych do opracowania i zweryfikowania stworzonych modeli i algorytmów zwiększa innowacyjność pracy.

Prognozowanie przepustowości i opóźnienia jest ważnym elementem opracowanego rozwiązania, które umożliwia pośrednie wymiarowanie pojemności stacji bazowej 5G (ang. BTS). W ramach badań zastosowano wielowymiarowe modele predykcyjne do prognozowania przepustowości i opóźnienia na poziomie plastra sieci jako podejście oparte na danych do wymiarowania pojemności 5G. Po kompleksowym porównaniu model VARMAX, czyli wektorowy model autoregresyjny średniej ruchomej z dodatkowymi zmiennymi egzogenicznymi, został wybrany jako najlepszy. Wyniki wskazują, że ten model jest równie

skuteczny w przypadku prognoz krótkoterminowych i długoterminowych z dobrą dokładnością. Ponadto włączenie wiedzy konfiguracyjnej, takiej jak pasmo częstotliwości, do procesu uczenia się modelu zwiększyło jego dokładność. Ocena jednowymiarowych modeli do prognozowania zmiennych środowiskowych została również przeprowadzona jako element wspierający dla modelu wielowymiarowego. Do tego problemu po dokładnej ocenie wybrano model Lag-Llama, który jest podstawowym modelem szeregów czasowych (ang. Foundational Time Series Model). Wszystkie walidacje i porównania przeprowadzono przy użyciu znormalizowanych metryk średniego błędu bezwzględnego i średniego procentowego błędu bezwzględnego.

Dodatkowo praca przedstawia oryginalną technikę wykorzystującą dane o ruchu na poziomie systemu do oszacowania statystycznego zysku multipleksowania agregowanych łączy transportowych 5G. Algorytm umożliwia skalowanie wyników symulacji. To podejście skraca czas obliczeń z dni do sekund, co ma kluczowe znaczenie dla rekomendacji dotyczących planowania sieci i ostatecznie zwiększa efektywność i elastyczność usług świadczonych operatorom telekomunikacyjnym. Zaprezentowano dwa studia przypadków, demonstrując zgodność oszacowań z wartościami zmierzonymi z łączy mikrofalowych w sieciach komórkowych i podkreślając ich znaczenie dla wymiarowania BTS w chmurze.

Wreszcie, wprowadzono oparte na danych rozwiązanie do prognozowania i wymiarowania, integrując koncepcję cyfrowego bliźniaka. Rozwiązanie to może autonomicznie służyć jako narzędzie prognostyczne do wymiarowania (plastrów) sieci i zarządzania ruchem, lub może działać jako kluczowy element kompleksowego cyfrowego bliźniaka. Ponadto ilustruje wykonalność tego, jak wzajemnie powiązane metody badane w tej pracy dostarczają niezbędne dane wyjściowe. Aby zweryfikować dokładność rozwiązania i ocenić jego przydatność, przeprowadzono eksperymenty na rzeczywistych danych. Wyniki pokazują, że proponowane rozwiązanie może skutecznie wyjaśnić i zmierzyć te zjawiska poprzez symulacje oparte na danych z plastrów sieci bezprzewodowych. Wdrożenie rozwiązania będzie niezawodnie wspierać procesy Nokii poprzez automatyczne rekomendowanie rozszerzeń pojemności lub konfigurowanie parametrów planowania podziału na podstawie rzeczywistych danych.

Contents

Chapter 1

Introduction

1.1 Motivation

The fifth generation of mobile technology introduces new service categories, of which the Enhanced Mobile Broadband (eMBB) represents an evolutional advancement of the Next Generation Mobile Networks (NGMN). However, 5G encompasses more than just increased speeds and capacity. The 5G architecture, characterized by reduced latency and improved availability, opens opportunities for the establishment of Ultra-Reliable, Low-Latency Communications (URLLC) and eMBB for critical applications [1], [2]. These applications introduce a range of new services that have the potential to revolutionize industries and improve our daily lives. However, these applications present a diverse set of requirements for 5G networks that must be addressed simultaneously.

5G technology is the latest generation of mobile networks available on the commercial market since 2019. This technology is groundbreaking in many ways and raises research challenges. The implementation of a Cloud Radio Access Network (Cloud-RAN) and the possibility of separating the radio protocol layers (functional splits) enable the introduction of the concept of distributed processing of radio signals (so far, the base station processing radio signals has been generally implemented in one physical device). This approach enables significant gains in the cost of network construction and maintenance for mobile network operators, thanks to traffic balancing and the use of "cheaper", shared computing resources in data centers for functions requiring greater computational complexity. However, the separation of radio protocol layers creates new requirements in the transport network between the elements of the distributed base station (fronthaul). That is, radio signals that were previously processed in one physical device must now be sent between separated elements in a limited time and with an increased transmission speed (than that resulting from the requirements of services provided to end users).

Network Slicing (NS) that was available in limited fashion in 4G is gaining momentum and full capabilities in 5G and emerging as a solution to meet the new range of needs. NS serves as a virtualization technique for the network that is parallel to the cloudification of the network function [3]. Each slice functions as a logical network built on top of the physical network and is equipped with the necessary resources to meet the specific demands of connected applications and users. The allocation of physical network resources to the slices can be shared or dedicated, and the dynamic assignment of slice resources enhances network efficiency and scalability. However, while NS offers numerous benefits, it also poses additional management challenges. Nokia Bell Labs estimates that the increased complexity of manual NS implementation could raise the total cost of ownership by 30 percent compared to traditional networks [4]. In contrast, the same research states that complete automation of NS could result in a 32% cost reduction.

The duties of Communication Service Providers (CSPs) engaged in the development and operation of NGMN include efficiently organizing, implementing and overseeing networks comprising multiple Base Transceiver Station (BTS). As networks grow, CSPs need to continuously evaluate and enhance their capacity. In cases of decreasing capacity, they initiate a network (re)planning process involving capacity dimensioning [2]. Typically, this sizing process is carried out using tools specific to the vendor, leveraging product capabilities and internal knowledge. The Nokia department called Network and Performance Engineering, the host of the author of this dissertation, is responsible for establishing the method and tools for this process. Therefore, the research carried out in the context of this doctoral dissertation contributes to the advancement of techniques for dimensioning 5G networks, with a specific focus on incorporating the impacts of Network Slicing (NS).

The increased challenge of sizing 5G networks compared to 4G and previous technologies stems from various factors associated with this innovative technology, as outlined below. 5G technology enables new services that have not been possible before. Nokia forecasts that the requirements for the 5G network that will enable the offering of these services will be 100 times higher than the current requirements for the 4G network [5], i.e. delays of 1 ms, peak rates of 10 Gbps, number of connected devices 100+ billion (where the current requirements for the 4G network are respectively, 100 ms, 100 Mbps, 10 billion).

In addition to the challenges associated with the dimensioning itself, there is a fundamental issue concerning its inputs. In order to accurately determine the necessary capacity, it is essential for the Communication Service Provider (CSP) or vendor to have detailed information on the expected traffic patterns, including the various services, their data volumes and the specific Quality of Service (QoS) requirements at a particular point in the future, ranging from a few weeks to several months, depending on the deployment schedule. While CSPs market penetration plans may suffice for long-term planning purposes, they may not be adequate for short-term forecasts or for scenarios involving time-varying wireless networks at specific locations [1]. The precision of the forecasts is related to the granularity of the product (which may differ per vendor). The capacity of the BTS increases incrementally with the addition of new hardware, each step expanding the capacity by several dozen percent up to an order of magnitude. The goal of dimensioning is to achieve accuracy below the smallest step, which ranges from a few to a dozen percent.

1.2 5G Technology Introduction

5G offers substantial enhancements in network capacity, connectivity, latency, and reliability. As illustrated in Fig. 1.1, this advancement enables a multitude of new services. The eMBB, which builds on existing mobile data services by greatly improving performance for high bandwidth requirements. It facilitates new business services and applications, as well as the booming use of multimedia and collaborative work environments, including Augmented Reality (AR)/Virtual Reality (VR) and various forms of video content. These frequent, collaborative and interactive communications occur not only between people, but also between smart devices, generating thousands of terabytes of data each day. The growing volume of mobile traffic generated by consumers demands greater capacity and reduced latency. With 5G, an anticipated peak data rate exceeding 10 Gbps will be achievable, a significant increase from the 450 Mbps provided by Long-Term Evolution (LTE). Furthermore, 5G aims for nearly zero latency, under 1 ms, ensuring that the radio interface remains efficient even for the most demanding applications.

The impressive connectivity and scalability provided by 5G enables the development of smart homes, smart cities, and smart factories, each equipped with billions of sensors that need a flexible and scalable infrastructure known as massive Machine Type Communications (mMTC). URLLC refer to crucial machine communications requiring exceptional reliability and minimal delay. The second phase of 5G, known as Rel-16 and standardized since 2020, emphasizes comprehensive support for the Industrial Internet of Things (IIoT) within Industry 4.0. This includes advanced URLLC and Time-Sensitive Communication (TSC), support for Non-Public Networks (NPNs), operation in unlicensed spectrum, and deployment improvements through Integrated Access and Backhaul (IAB) operation, focusing primarily on mmWave networks. From the viewpoint of the 5G System Architecture, Rel-17 and subsequent releases offer (among other things) improved

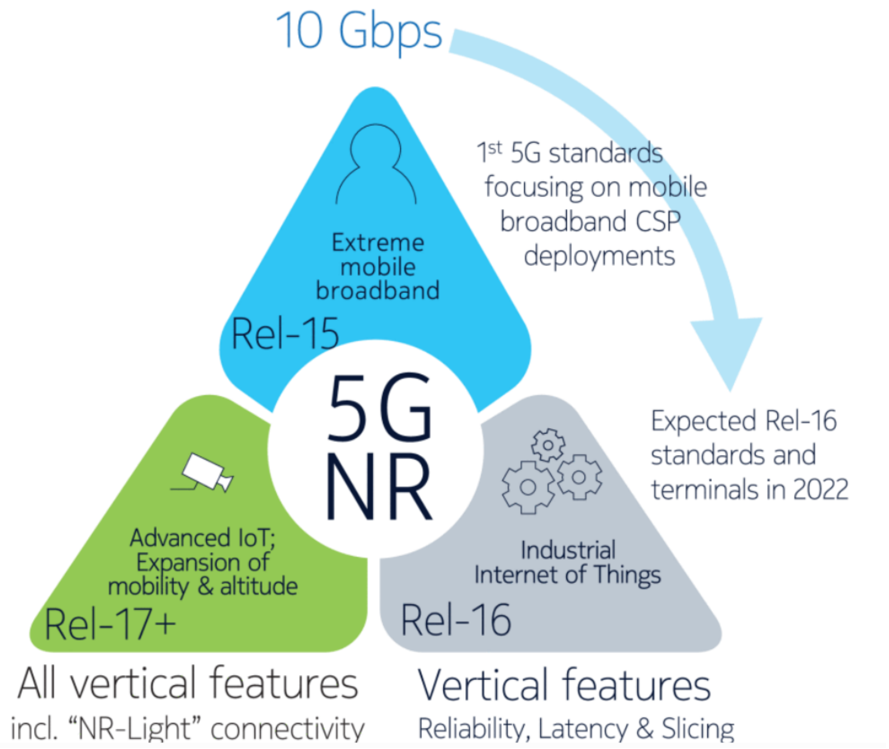

10 Gbps
1st 5G standards focusing on mobile broadband CSP deployments
Extreme mobile broadband
Rel-15
5G NR
Expected Rel-16 standards and terminals in 2022
Advanced IoT; Expansion of mobility & altitude
Rel-17+
Industrial Internet of Things
Rel-16
All vertical features
incl. "NR-Light" connectivity
Vertical features
Reliability, Latency & Slicing

Figure 1.1: *Evolution of 5G from Rel-15 to Rel-17 (source: [6]).*

support for IIoT, augmented support for NPN, better support for the convergence of wireless and wireline networks, multicast and broadcast architecture support, proximity services, enhanced multi-access edge computing, and increased support for network automation.

NS is a solution to accommodate described wide range of demanding requirements for latency, throughput, capacity, and availability. NS creates comprehensive logical networks that possess isolated properties and operate independently. As new services are added to the network, a cloud-native core is capable of generating an instance, or slice, of a complete network virtually. This slice is thoroughly tailored with network resources (dedicated if necessary) assigned by use case, subscriber type, or application from a unified infrastructure. NS provides an efficient method to satisfy the needs of numerous services and applications over a shared network infrastructure, including smartphones, tablets, VR, personal health devices, essential remote control equipment, and automotive connectivity.

1.3 Important Challenges for 5G Network Dimensioning

The dimensioning process is commonly carried out using vendor-specific tools that make use of the capabilities of the product and the internal expertise. Historically, this has been a manual process that is heavily based on spreadsheets. Furthermore, the approach typically follows standard linear or queueing models [7], which are refined and confirmed through network simulations or laboratory experiments. Incorporation of product improvements, such as new functionalities or expansions, is included in the model as an extra linear element, which can result in inaccurate results in intricate deployments or with the introduction of NS due to multicollinearity issues [8].

1.3.1 One-Fits-All Process

One component of the Radio Access Network (RAN) planning procedure involves capacity dimensioning [2]. The purpose is to calculate the amount of resources needed to provide a service for a particular traffic volume while maintaining appropriate QoS. Typically, this process comprises the stages illustrated in Fig. 1.2, which can be carried out sequentially or concurrently.

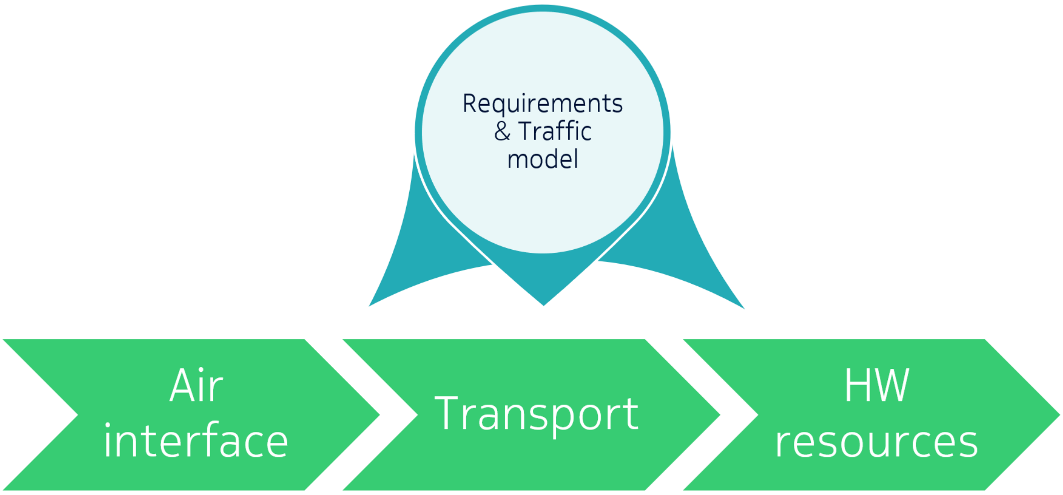


Figure 1.2: *General BTS dimensioning process flow.*

Initially, it is necessary to perform an air interface dimensioning to roughly estimate site volumes during the implementation of a radio network. Following this, transport dimensioning is carried out to project the necessary capacities on the access network interfaces [9]. Lastly, Hardware (HW) resources dimensioning is performed to determine the quantity of physical resources needed, which are measured in modules (such as radio or system) for traditional bare metal installations and in Central Processing Unit (CPU) for cloud-based deployments.

All these phases necessitate the specification of common inputs, such as network configuration (e.g., cell bandwidth, Multiple-Input Multiple-Output (MIMO) mode, functional split) and traffic model (e.g., number of subscribers, data volume), which are contingent on the CSP requirements (derived from the technology vision). The dimensioning process is typically performed using vendor-specific tools, as the specific constraints of the equipment are proprietary to the vendor. Currently, this process often relies on spreadsheets and lacks a more advanced method [8]. In addition, planning engineers must have experience performing traffic forecasting and performance estimation. Manual or so-called "spreadsheet" planning complicates the dimensioning of the entire network with diverse configurations and varied traffic models and services, making it challenging or even unachievable. This method is inefficient and promotes a "one-rule-fits-all" approach that assumes uniformity across all cells. In addition, manual planning is susceptible to human errors, which can lead to inaccuracies.

The distinction between two scenarios is important for dimensioning. A long-term scenario is necessary when a CSP is considering Capital Expenditures (CapEx) investments (typically for 1-2 years) and is in the process of selecting a vendor(s) to supply equipment for network deployment. In this situation, the tools used should forecast adequate resource planning for future requirements. On the other hand, a short-term scenario is required for the daily operation of the network with slicing, as all the slices compete for the limited resources (previously estimated in the long-term scenario) based on immediate needs. Although both scenarios produce the same outputs (site capacities), they may require different approaches due to the varying time scales involved.

1.3.2 *A Priori* Defined Traffic Model

The dimensioning process involves gathering input data that describe the operating conditions of the network and the services that it is expected to provide. The Traffic Model (TM) serves as a numerical representation of this. Forecasts of traffic data are used to establish the TM for a future period when the network is scheduled to be operational [10]. CSP supplies estimates of the number of subscribers it anticipates having on its network over the next few years. Information on the types of services and the demand for each service per subscriber is also given, typically derived from a generic statistical model that is linearly extrapolated over time. Although this approach may suffice for long-term planning, it may not offer precise results for specific site requirements or short-term forecasts in dynamic wireless networks (particularly in high-mobility network slices) [1]. Even with highly accurate dimensioning tools, the accuracy of the results is dependent on the quality of the TM [1]. Hence, it is crucial to enhance the dimensioning process by incorporating traffic forecasting techniques based on actual network data or by designing the network without relying on the TM [11].

1.3.3 Linear Models

Dimensioning models for mobile networks are typically developed through network simulations during the product development phase and are refined with test results upon completion. Due to time and resource constraints, simulations and tests are generally conducted for only a subset of potential scenarios that involve various network configurations, new functionality activations, and traffic mixes. Consequently, standard linear or queueing models are formulated to align with the simulated and tested scenarios [7]. Any unexplored or untested scenario, particularly those with intricate configurations and feature combinations, can compromise the accuracy of the dimensioning model. The introduction of network slicing further complicates the process by multiplying the traffic mix and associated service requirements for each specific scenario. The impacts of individual product features are incorporated into the model as an additional linear model or coefficient. It is important to note that not all features act independently, and they often influence each other's performance. Consequently, the concept of "feature bundles" needs to be either simulated or tested, which adds complexity to the development of the dimensioning model.

Simply adding coefficients or using feature-specific linear models may lead to inaccurate results due to multicollinearity issues [8]. Although using standard linear models in the dimensioning process can provide precise results for specific scenarios, the reliability of the process decreases as the number of configurations and features increases. These insights have prompted investigations of nonlinear modeling approaches, including Artificial Intelligence (AI) driven techniques. Furthermore, employing traditional model-based iterative dimensioning methods may not be suitable, as the computational complexity increases significantly with the scale of the network [1].

1.3.4 Cloud Architecture Impact on Transport Dimensioning

The primary advantage of Network Functions Virtualization (NFV) is its ability to facilitate faster resource scaling compared to traditional Physical Network Functions (PNF). In PNF setups, the procurement, commissioning, and connection of HW were necessary before making it available for the deployment of a new network application. Virtual Network Functions (VNF) are then implemented as software applications on top of a Network Functions Virtualization Infrastructure (NFVI) to deliver telco services on the operator's premises [1]. Cloud-RAN allows for the division of the BTS into Radio Unit (RU), Distributed Unit (DU), and Centralized Unit (CU) that can be deployed at different locations (Fig. 1.3), to allow resource sharing. In contrast, in traditional setups, fronthaul and midhaul connections are typically established using physical point-to-point links, since all components are co-located. This requires that CSP take into account capacity planning for fronthaul and midhaul links in addition to backhaul. In classical RAN deployments, the final stage of the capacity planning process (Fig. 1.2) is uncomplicated because the Software (SW) is closely tied to the HW, making the capacities of SW+HW known throughout all planning stages. However, for cloud-based planning, an extra step is required to map virtual resources to physical hardware resources.

1.3.5 Access Transport Aggregation

The data carried over the radio interface are transmitted through the access transport network to and from the components of the core network, establishing aggregation points (Fig. 1.3). Transport connections must have adequate capacity and QoS to support the necessary radio operations. As a result, a key aspect of RAN planning involves estimating the capacity of individual BTS interfaces and aggregation points within the access domain. It is important to note that the variability in packet traffic presents an opportunity for cost savings through statistical Multiplexing Gain (MG) [9].

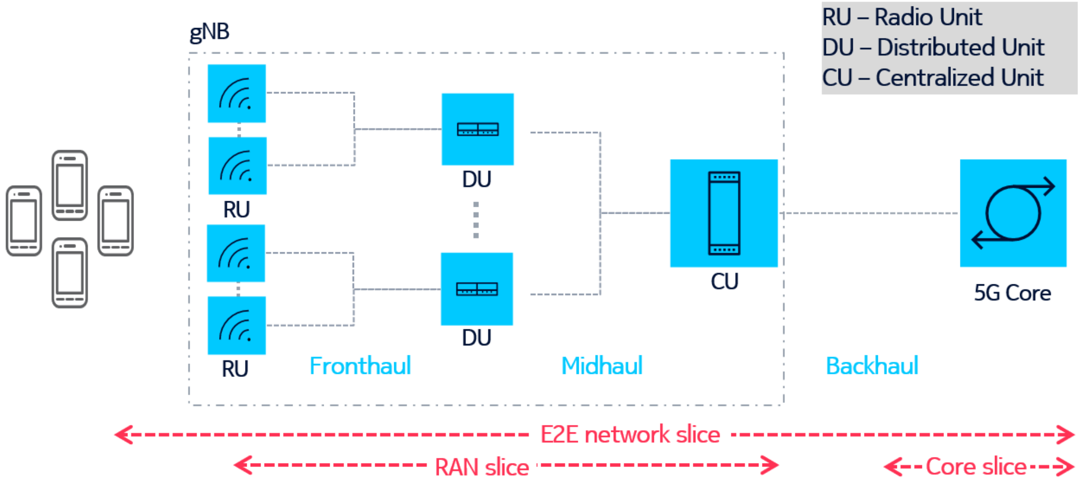


Figure 1.3: *5G cloudified gNB with transport interfaces and network slices.*

Once again, to understand the QoS needs (e.g. temporal throughput values) of the radio interface and traffic profile patterns for specific scenarios, it is necessary to conduct simulations at the system or network level following the traditional dimensioning approach. These activities form the basis for creating linear models for each scenario, assuming a uniform network configuration and traffic demands. This presents an opportunity for potential improvements. Furthermore, cloudification and division of BTS functions impact transport, particularly fronthaul, depending on the functional split between RU and DU [12]. In the current phase of 5G deployment, the capacity of the fronthaul link is determined by the split in a static manner. In the subsequent phase, as the fronthaul functionalities evolve and the need to reduce its capacity arises, it will rely on the traffic it transmits.

Transport planning plays a crucial role in the comprehensive planning of the RAN since optimizing the costs of the transport network can result in general cost savings in the deployment of the mobile network. It is essential that transport connections offer adequate capacity and maintain a high QoS to support the necessary radio performance, which is typically included in the service agreements and marketing strategies of CSP. Consequently, while there is a need to reduce costs in the transport network, this should not compromise the performance of the radio interface.

1.3.6 Network Slicing

The introduction of network slicing adds an additional layer of complexity to the challenges mentioned above. Since each slice may have unique requirements such as Traffic Management and QoS, and the network resources are shared, all these requirements must be considered when planning. The network slice, which is divided into RAN slice, core slice, and transport slices [12] (refer to Fig. 1.3), further complicates the dimensioning process. Although slice requirements are specified end-to-end (Fig. 1.4), each slice component is evaluated separately due to the specific characteristics of the underlying resources, such as radio, transport, and core. Consequently, each slice component may require a dedicated model.

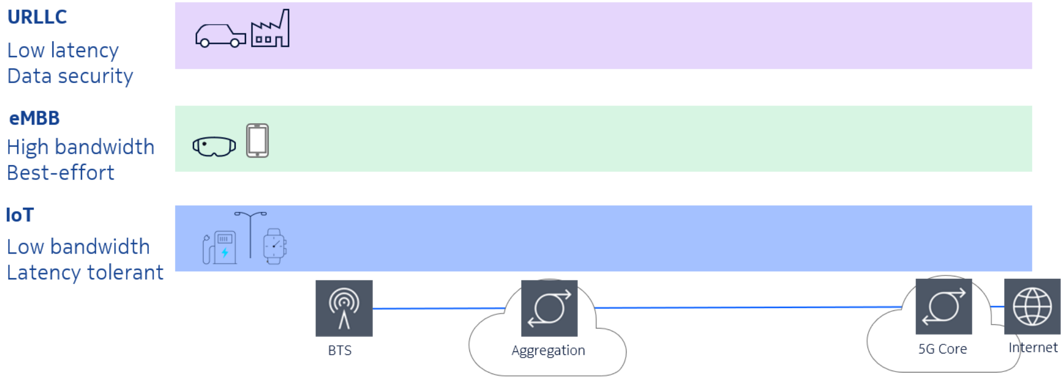


Figure 1.4: *5G Network Slicing conceptual architecture.*

Another area that raises many questions in the domain of network dimensioning is the new distributed architecture of the access network based on cloud solutions. Moving from the previously monolithic (apart from the antennas themselves) construction of base stations to several geographically separated elements creates the need to replan the transport network. Additionally, the implementation of virtual network slices will complicate this process because it will add an additional dimension (several logical networks within one physical one). The above changes generate the following research questions:

- What is the impact of different functional splits on the requirements in the transport network (fronthaul)?
- What are the benefits of aggregating multiple radio signal processing elements in one place/cloud in terms of transport network capacity?
- Is it possible to dimension one 5G base station (as it was in 4G), or does it have to be done for the entire part of the access network that is aggregated in the same cloud?
- What is the impact of virtual network slices on the transport network?

The current method for dimensioning the 4G transport network focuses on the part between the base station and the elements of the backbone network. It is assumed that due to the increase in the complexity of the architecture for 5G, it will be necessary to develop a new method for dimensioning the transport network, which will have to take into account its multidimensionality (new services, distributed base station in the cloud, virtual slices).

1.4 Contributions

The research hypothesis of the doctoral dissertation is as follows.

It is possible to improve the best-in-practice 5G network dimensioning procedure using data-driven modeling of the forecasted throughput and delay in network slices.

In order to prove the research hypothesis, the following goals are formulated:

- Collection and analysis of 5G BTS network data.
- Selection of methods and their parameters for verification.
- Development and verification of one-dimensional method for short- and long-term environmental variables forecasting.
- Development and verification of multi-dimensional method for short- and long-term throughput and delay forecasting.
- Development and verification of method for multi-cell traffic MG estimation.

- Development of network slicing dimensioning framework enabling simulating "what-if" scenarios.
- Verification of developed methods in industrial use cases.

AI methods, particularly Machine Learning (ML), help to leverage a wider range of input data to improve the accuracy of traffic model forecasts, which is a key input condition for dimensioning. Additionally, created methods enable the sizing of multi-access mobile network slices, optimizing the use of network resources. Moreover, thanks to the limitation of human input, this research contributes to enabling data-driven decision-making and improves the reliability of the dimensioning procedure.

The results of the doctoral dissertation and associated research contribute to the advancement of the 5G network and will be utilized in Nokia's network planning and dimensioning processes. Specifically, the NS dimensioning framework will be incorporated into the 5G network dimensioning tool used in the aforementioned processes. A portion of the work has already been completed, as detailed in the corresponding sections. It is believed that the previous, simplified dimensioning procedure, which considered manually prepared traffic model, resulted in the over-allocation of network resources. The new data-driven framework that considers the slice-level requirements will limit that.

As part of the research, ML methods are employed to address the issues encountered during the dimensioning of the 5G mobile network technology, its functionalities (NS), and the services it offers. Numerous computer experiments were conducted using a variety of datasets (e.g. live commercial networks and simulators) to identify suitable ML algorithms and determine their parameters (tuning the methods to the problem). This approach is deemed appropriate due to the increased complexity of the 5G network (e.g., virtual slices, Cloud-RAN), which heightens the difficulty of solving the dimensioning problem, potentially rendering current methods inadequate. Furthermore, this approach is innovative as it is not yet utilized in the scientific literature. The use of real network data to develop and verify the methodologies and algorithms created adds to this innovation. Additionally, since the network dimensioning process consists of multiple steps, research will require careful selection or adaptation of statistical and artificial intelligence methods for each step. For example, preparing input data that describe network traffic (planned services provided) is one such step. These data must be inferred from current information or forecasted for the future. Statistical tools can be used for this purpose, but it can be challenging to account for all factors affecting traffic (e.g., seasonality, daily traffic variations, growth in the number of 5G devices and users). Thus, research and data analysis will require the selection of appropriate methods to address problems at each step of the overall process.

1.5 Dissertation Structure

Chapter 2 covers the evolution of 5G networks, offering a summary of 5G traffic prediction, NS dimensioning, and the current methods for estimating transport aggregation MG. Furthermore, it introduces the notion of Digital Twin (DT), which serves as the foundation for this research. Chapter 3 describes the essential characteristics of the real 5G network data utilized in this study, highlighting the key aspects. In addition, it discusses the challenges encountered when dealing with this type of data. A thorough literature review led to the identification of the methods that are evaluated in this doctoral dissertation, detailed in Chapter 4. Chapter 5 discusses the selection process and the criteria for the models used in forecasting environmental variables. Subsequently, Chapters 6 and 7 address the comparison of models for short-term and long-term forecasting, respectively. They also provide results and discussion of the selection of models for the final solution. Chapter 8 discusses the statistical aggregation gain seen in RAN and the algorithm designed to estimate it. Chapter 9 illustrates the interconnections of all the models, methods, and algorithms introduced earlier, forming the 5G NS dimensioning framework. Furthermore, it explains its application in scenario simulations that integrate the DT concept. Chapter 10 wraps up the study and outlines future perspectives.

Chapter 2

Related Works

This chapter outlines the current advancements, concentrating on the primary research challenges tackled in this dissertation, such as NS dimensioning, 5G traffic forecasting, estimation of benefits of transport aggregation, and development of a DT.

2.1 Network Slicing Dimensioning

In terms of research work related to network slice planning, the following publications are worth mentioning. A description of the 5G network cell planning process, the challenges it faces, and an overview of current approaches to solving them is included in [13]. In [14], the authors try to solve the problem of maximizing the profit of a 5G Cloud-RAN operator by appropriately accepting requests to create new network slices. The scope of work includes two main services of the 5G network: eMBB and URLLC. The allocation of spectrum and other base station resources to network slices is the subject of the solution developed in [15]. Another important aspect of network dimensioning is the development of a TM. In [16], the authors describe the challenges for planning and dimensioning 5G networks with NS, the solution of which will require the use of AI techniques. One of the significant problems described in this work is the need to move away from predefined TM to methods that allow its forecasting. A solution for predicting 4G network traffic using Markov chains is described in [17]. In [18], the authors describe a method to forecast several network performance parameters using ML. In [10], the authors deal with the problem of forecasting the telecommunications traffic measures in the next time window, based on previous observations, using neural networks. However, the proposed model is designed to be used for short time windows, that is, a few seconds, so this method can be used for traffic balancing purposes, but not for forecasting for dimensioning purposes (where the time window is months). In [19], the authors developed a neural network architecture that can accurately forecast traffic 10 hours in advance based on network data. [20] presents an algorithm that utilizes the alternating direction method of multipliers to distribute processing and bandwidth resources across slices.

In [2], the authors developed a model to dimension various services in the 5G network using real network data based on heuristics. The purpose of the model is to estimate the network resources that implement the radio interface in terms of capacity and range while ensuring the defined QoS. In [21], the researchers proposed a model to dimension the fronthaul of the 5G network to guarantee minimal delays for URLLC. They used the G/G/1 queueing model. A similar approach to structure the transport network (specifically LTE) using a queueing model was discussed in [22], where they used a Poisson model with Markov modulation MMPP(2)/D/1.

Extensive research has been conducted on the utilization of AI-based solutions for NS management, which can be applied in all stages of network management (preparation, planning and operation) [7]. These solutions also show promise in addressing complex decision-making challenges within dynamic network settings, such as optimizing transmission power in cellular networks and managing resource allocation in network slices. In [1], the authors demonstrate that ML algorithms facilitate the modeling of individual cells according to their unique characteristics, allowing the planning of heterogeneous networks while taking into account local requirements. Another example involves utilizing game theory for the allocation of slices in the RAN planning process [23]. A supervised deep neural network is suggested for the allocation of spectrum, with the goal of reducing costs, optimizing the utilization of radio resources, and ensuring the fulfillment of desired service level agreements as described in [24]. In [25], a dynamic slice reconfiguration framework is introduced. This framework facilitates vertical and horizontal scaling operations to manage time-varying loads. The proposed solution utilizes a mixed integer quadratically constrained programming method and has been validated through simulations.

Current research is devoid of studies that illustrate how to distribute network capacity based on actual network data to meet slice QoS requirements.

2.2 5G Traffic Forecasting

Network traffic forecasting can be performed using offline or online methods [26]. Offline approaches gather data on the entire time series before making forecasts, while online methods focus on specific data segments and update model parameters sequentially based on new information. Currently, numerous studies on 5G network dimensioning leverage

ML techniques. Among these, neural networks with Long Short-Term Memory (LSTM) units are widely used [26, 27].

The study carried out by the authors in [28] validates that LSTM exceeds the performance of alternative methods via the Diebold and Mariano test. This test evaluates if the prediction quality of the i-th method is inferior compared to the j-th method. The N-Beats model [29] is a multi-branch neural network framework designed for forecasting time series data, especially effective with multidimensional datasets, and is applied for aggregated traffic prediction in [30]. An alternative instance where fundamental neural network architectures, such as Convolutional Neural Network (CNN), dense, and LSTM networks, are used for the prediction of network traffic is documented in [31], focusing on a 24-hour forecast. The dataset encompasses not only traffic data but also weather, electricity consumption statistics and location, which is a unique feature of this study. The authors emphasize the significance of runtime as a critical aspect in algorithm evaluation, as it is prudent to consider this when assessing models. This is because time can be a decisive element for the practical applicability of a method.

When training a neural network or any ML model, selecting an appropriate loss function is crucial. In [32], the authors introduce the DeepCog method, a deep learning strategy utilizing an encoder-decoder architecture paired with an asymmetric loss function. DeepCog translates any type of traffic described within the network slice into a tensor form. Its encoder-decoder framework facilitates the prediction of future throughput. The loss function proposed addresses the challenge of balancing the over-allocation of resources with maintaining service level agreements. Due to its universal framework, DeepCog can be employed to manage various levels of traffic aggregation. When employing LSTM-based neural networks, numerous researchers opt for one-step-ahead forecasting, which is not universally appropriate. A comparison of one-step versus multiple-step-ahead predictions is detailed in [33–35]. While neural networks are a powerful tool across various applications, developing an effective framework and training them for time series prediction presents considerable challenges.

One limitation of neural network approaches is their inability to provide probabilistic uncertainty quantification [36]. In contrast, statistical methods offer a different scenario. The concept of predicting traffic through time series has been effectively applied for years, including with earlier telecommunications network generations.

Among the widely favored models for forecasting network traffic using univariate time-series are the Autoregressive Integrated Moving Average (ARIMA) models [37–40] and Exponential Smoothing methods, including their advanced variants [40–42]. Several studies deploying these time-series models focus exclusively on the univariate scenario, although some do acknowledge the potential for analogous analyses involving multivariate datasets.

Several articles also explore methods for refining exponential smoothing techniques and their advanced formulations [41]. Another approach discussed in [42] involves comparing the outcomes of ARIMA and exponential smoothing models. One study demonstrates that the ARIMA model is less precise for predicting single-cell throughput. However, ARIMA performs better for forecasting throughput on weekdays when considering an entire region within an LTE network. It is important to note that most studies employing time series models focus exclusively on univariate cases. Some of these studies suggest the potential for conducting similar analyses with multivariate data. Multidimensional models are rarely utilized due to the complexity of selecting suitable predictor variables for traffic forecasting or due to challenges in accessing authentic multivariate data. In [36], the authors compare the performance of seasonal ARIMA with a Gaussian Process (GP) approach using real 4G BTS data. Additionally, they introduced a feature embedding kernel specifically designed for a GP model to predict traffic demand, enhancing peak-trough accuracy compared to overall accuracy.

Various studies have assessed the precision of intelligent techniques in conjunction with time series models [28, 43]. For example, the work in [44] presents a comparative analysis of several methods including ARIMA, Support Vector Machines (SVM), the historical average model, Fusion Prior Knowledge Network (FPK-Net), LSTM, Transformer, and a newly proposed neural network architecture (incorporating an LSTM block, a convolution layer, and an Attention module). Researchers often opt to combine time series and deep learning techniques to take advantage of both. The study in [45] details a technique that employs Thresholded Exponential Smoothing (TES) and Recurrent Neural Network (RNN) for predicting allocation of network resources and mobile traffic anomalies. It highlights the benefits of imposing penalties related to Service Level Agreements (SLA) breaches. It is also effective for anomaly detection, offering a speed advantage over Bayesian methods. However, its limitation lies in the effectiveness of exponential smoothing primarily for one-dimensional data. Another example of integrating AI with time series analysis for traffic forecasting is outlined in [46], where Discrete Wavelet Transform (DWT) is used for time series decomposition, followed by modeling a linear component with ARIMA and predicting a non-linear component with LSTM.

Researchers employ various techniques to predict network traffic, such as classification approaches, methods based on information theory, (hidden) Markov models, Gaussian processes, and Poisson models [27, 32, 36**?**]. In the academic works, there are also studies that utilize supervised techniques for network traffic prediction [26, 47–49]. These approaches encompass support vector machines, k-nearest neighbors, decision trees, linear regression, AdaBoost, and random forest. A drawback of these techniques is the need to manually pre-process the dataset to indicate weekly patterns. However, incorporating numerous shifted values for each variable can lead to a curse of dimensionality, meaning that increased dimensionality leads to data sparsity. It is important to consider periodicity at both the daily and weekly levels, as network congestion exhibits a seasonal pattern. For instance, thesis [27] illustrates how average traffic values fluctuate at different times of the day. In addition, the author highlights the variation in intraday traffic volatility between weekend data and weekday data. Recognizing these patterns can improve the accuracy of the forecast.

The use of advanced statistical or ML models that take advantage of data allows detailed modeling of individual cells or groups of cells with similar configurations or performance traits, tailored to their distinct features [1], [50]. Models that incorporate multiple variables take into account a variety of factors to forecast traffic growth for each cell. This method results in precise and diverse resource planning for networks, addressing specific local needs rather than using a uniform approach. The traffic model is essential for dimensioning. Recent studies show that AI methods such as Deep Neural Network (DNN), LSTM, and RNN can accurately forecast traffic load specific to services [1].

The use of real network data allows for the application of forecasting techniques to accurately predict future resource requirements. This challenge can be divided into short- and long-term forecasts. Short-term forecasts improve operational management and can be advantageous in network management, where continuous predictions are used to optimize specific processes (e.g., capacity allocation per slice). Short-term forecasting is particularly useful in environments with high variability. An example of a short-term prediction for 5G data is a power forecast for 5G photovoltaic base stations [51]. In this scenario, the aim is to achieve more accurate energy allocation and management. The authors state that short-term energy fluctuations are affected by low latency, high data transmission rates, and the diversity of connected devices specific to 5G. The benefit of long-term forecasts is the ability to plan future strategies over an extended period. However, their limitation observed in systems is the chaotic dynamics of these systems [52].

Recent advances in generative AI models have significantly impacted the field of time series forecasting, prompting extensive research into how these models can be utilized. Foundation Time Series Models (FTSMs) are a category of ML models trained through self-supervised learning (the model learns to identify patterns and relationships within the data without needing explicit instructions or labeled data) on vast and diverse datasets. These models can then be fine-tuned for numerous downstream tasks. The concept of pre-training models on initial tasks and subsequently fine-tuning them for specialized tasks is referred to as transfer learning. The aim of transfer learning is to leverage the knowledge obtained from pre-training on a broad dataset to improve the performance of a similar or the same model on a distinct, more specific task, or on a smaller dataset. Recent progress in this field has transformed the approach to model design in time series analysis, improving various downstream applications [53–58].

Among the diverse range of models, the following are identified as having the highest potential to address certain challenges in this doctoral dissertation. The Lag-Llama model stands out as a foundational model for univariate probabilistic time series forecasting, founded on a straightforward decoder-only transformer architecture [59]. Lag-Llama is designed to generate a probability distribution for each predicted timestep. Although it demonstrates strong zero-shot capabilities (to perform predictions on fresh time series data without the need for retraining or adjusting the architecture's weights), its performance improves significantly with fine-tuning [60]. Performance enhancement is directly proportional to the amount of data used for fine-tuning. The Chronos model utilizes a method in which a time series is converted into a series of tokens through scaling and quantization. These tokens are then used to train a language model with the cross-entropy loss function. After training, probabilistic forecasts are generated by sampling several potential future paths based on historical data [61]. The Chronos model has been trained on an extensive dataset of publicly available time series and synthetic data created with Gaussian processes. TimesFM is a forecasting model, initially trained on an extensive time series dataset comprising 100 billion real-world time points, showcasing remarkable zero-shot performance across diverse public benchmarks spanning multiple domains and levels of detail. This model is specifically a decoder-only foundation model tailored for time series forecasting. Compared to the most recent Large Language Models (LLMs) (e.g. GPT-3.5's architecture comprising of 175 billion parameters), TimesFM is significantly smaller, with only 200 million parameters [57].

Despite extensive research, there is no publicly accessible application that demonstrates the use of FTSM in the telecommunications sector.

This doctoral dissertation aims to propose a comprehensive framework for network traffic forecasting that takes into account seasonal data variations. Specifically, the framework should integrate the interplay between Key Performance Indicators (KPIs), e.g. throughput and delay throughout all the network sections considered. A review of the current literature highlights the potential for further research in this field.

2.3 Estimation of Transport Aggregation Gain

To address the demanding requirements of the fronthaul network, numerous research studies have been conducted. The findings of the study in [62] demonstrate significant bandwidth conservation for variable bit rate systems due to statistical MG. The publication in [63] explores various optical transport network structures and technologies that can be used to construct an effective fronthaul network for 5G. Using queueing theory and spatial traffic models, the research in [64] calculates the statistical MG achieved by adjusting the number of user streams, demonstrating a substantial reduction in fronthaul capacity required based on traffic demand and statistical characteristics. In [65], the authors introduced an accessible model to quantitatively assess the statistical MG of the fronthaul. They achieved this by deriving the probability of user blocking resulting from the restricted fronthaul capacity, as well as calculating its upper and lower bounds. Subsequently, a large limit analysis was employed to derive the closed-form expression of the fronthaul statistical MG, facilitating the quantification of the gain for substantial cluster sizes.

The use of the statistical MG is leveraged in [66] to distribute surplus resources from over-served slices to under-served slices, taking into account the actual channel conditions of the associated user equipments. An assessment of computational and power savings facilitated by Cloud-RAN is presented in [67] through a quantitative analysis considering various RAN functional splits and using a multidimensional Markov model. Likewise, a multidimensional Markov model is employed in [68] to assess the statistical MG of virtual base station pools. Another study [69] focuses on offering recommendations for network deployment. The authors introduce an equation that computes MG by integrating the spatial distribution of the data traffic, which is verified by simulations.

While it is a widely-studied subject within the transport network field, there is no work that demonstrates how the MG is influenced by traffic type and volume, nor how simulations or real network data can be utilized for scaling and estimation purposes.

2.4 Digital Twin

To promote the development of more efficient network management and tools for modern communication networks, the concept of DT was proposed. These tools encompass troubleshooting, traffic engineering, "what-if" scenarios, network planning, and anomaly detection, as depicted in Fig. 2.1 from [70]. The study also highlighted how advancements in ML facilitate the creation of crucial components of network DT through data-driven network models that can operate in real-time, such as routing optimization within a

QoS-aware context. [71] introduced a novel framework for a DT manager designed to handle conflicting network applications and devised a DT model for "what if" analyses to optimize the border gateway protocol. Furthermore, [72] went into softwarization and intelligentization of 5G/6G networks, underlining the importance of the DT architecture for network autonomy. The authors foresee that a service layer in 6G networks will emerge, aligning with DT technology and incorporating proactive analytics, including generative intelligence features.

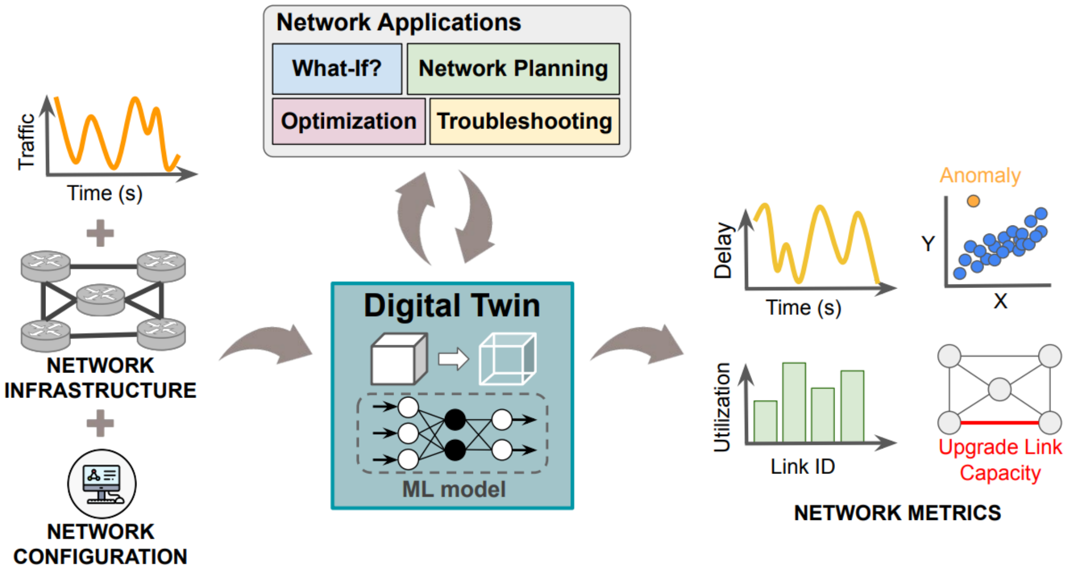


Figure 2.1: *General network digital twin architecture (Source: [70]).*

Chapter 3

5G Network Data

3.1 Dataset

The study used a dataset comprising hourly averaged time series data from thirty three 5G BTS operating in a live network deployment. Data were collected from each BTS and its configured cells over the course of March 2023 for short-term and June 2023-February 2024 for long-term forecasting. The entire dataset has been divided into excerpts for both short-term and long-term forecasting based on the necessary model training duration. Whenever specific BTS or cell data is presented, it's IDs are given, e.g. BTS-1, CELL-1.

Each cell is characterized by various configuration parameters (such as cell duplex mode, channel bandwidth, etc.) and performance metrics (KPIs such as throughput). KPIs are calculated from counters, which provide detailed information on network events at a low level (e.g., the total downlink Radio Link Control delay in gNodeB DU per slice) [73].

Subscribers within this network cluster have been segmented into four distinct groups:

- Slice A - mobile subscribers with high priority
- Slice B - mobile subscribers with medium priority
- Slice C - mobile subscribers with low priority
- Slice D - fixed wireless access subscribers with lowest priority

Due to confidentiality and legal obligations, the data from the commercial networks presented in this dissertation have been anonymized and normalized. Normalization is performed by subtracting the minimum value and dividing by the range per each cell and slice separately. This approach was taken in a way that preserves the data's informational value and is referenced whenever relevant. Furthermore, details of the analyzed BTSs, including location, configuration specifics, operator, etc., cannot be shared.

Table 3.1: *The variables utilized in the modeling process.*

Abbreviation	**Full name**	**Unit**	**Description**
#UEs	Number of user equipments	#	The average number of user equipments which have buffered data in downlink direction
CQI	Channel Quality Indicator	#	This indicates the average level of modulation and coding the UE could operate
PRB utilization	5G PRB utilization for PDSCH	%	Utilization of PRBs for physical downlink shared channel (PDSCH)
BLER	Block Error Rate	%	A ratio of the number of erroneous blocks to the total number of transmitted blocks
DV	Data Volume	kbit	Amount of data send per particular network slice
TPut	Throughput	kbit/ms	Average downlink throughput volume at PDCP SDU level for a given network slice
-	Delay	microsecond	Calculated as time difference between the reception of the RLC SDU from PDCP layer and when first RLC SDU is sent over the air interface

3.2 Model Feature Selection

The KPIs used to assess the performance of 5G BTS have been carefully chosen based on the author's expertise in telecom network data, past analytical projects, and the evaluation of KPIs dependencies described in section 3.4. These selected KPIs are core performance indicators found in any vendor's radio equipment, facilitating the creation of multivariate models that incorporate both traffic load and radio environment metrics [74], which directly affect throughput and delay, as discussed in the following subsection.

The characteristics used in the modeling process are depicted in Fig. 3.1. Detailed information including the complete name, description, and unit for each variable can be found in Tab. 3.1.

The variables were divided in two areas:

- traffic conditions: #User Equipments (UEs) , Data Volume (DV), Physical Resource Block (PRB) utilization,
- environmental conditions: Channel Quality Indicator (CQI) and Block Error Rate (BLER).

The metrics indicated in Fig. 3.1 as *Total* are computed for all slices, while the metrics identified as *Slice* are computed individually for each slice. Exogenous and endogenous variables have been determined on the basis of the domain knowledge of the subject.

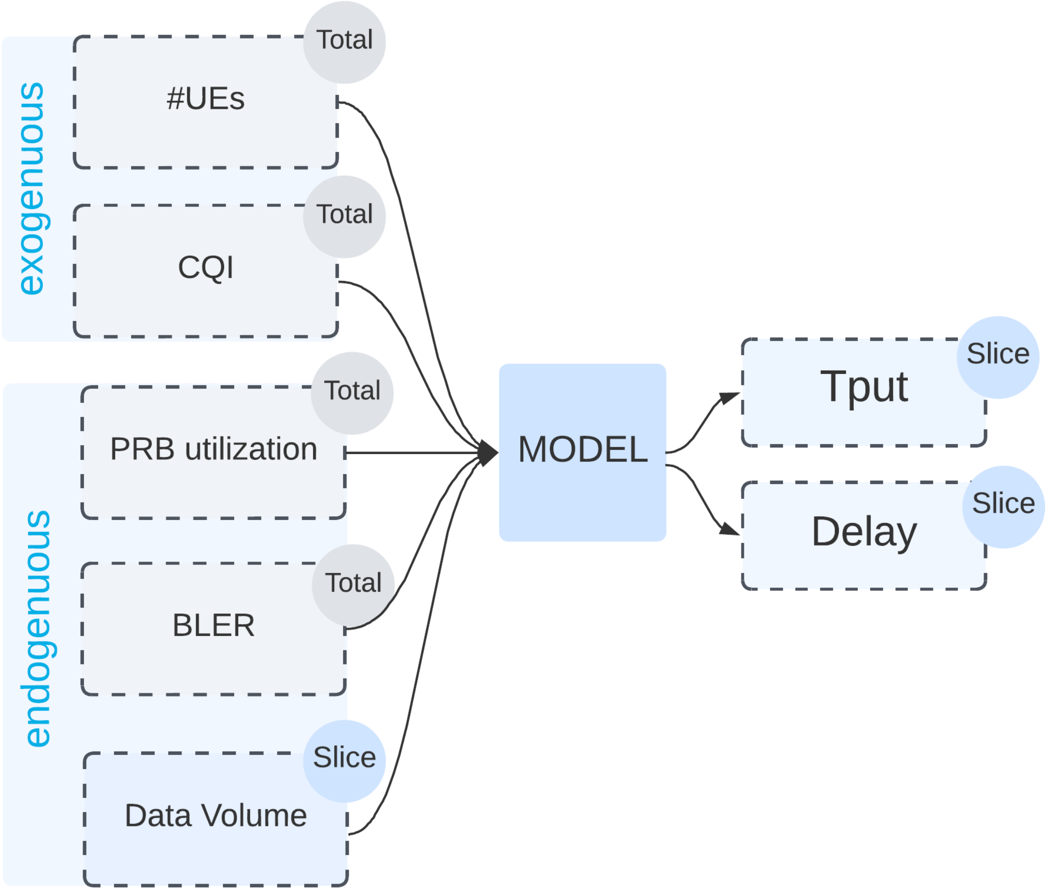


Figure 3.1: *Diagram of the model for predicting throughput and delay.*

UEs, CQI are identified as metrics determined externally from the model, while PRB utilization, BLER and DV are recognized as metrics influenced by the model.

3.3 Configuration Changes

Initial trajectory analyzes have indicated that certain time series exhibit alterations in their structure (Figs. 3.2 - 3.5) with three periods marking different signal characteristic. These changes are often attributed to adjustments made by the network operator. Consequently, it is crucial to gather configuration data that detail software modifications and feature activations. This information enables the identification of segments within the data that remain consistent. The study involves examining the dates of significant configuration modifications in the network to isolate unchanged segments that are later utilized for modeling.

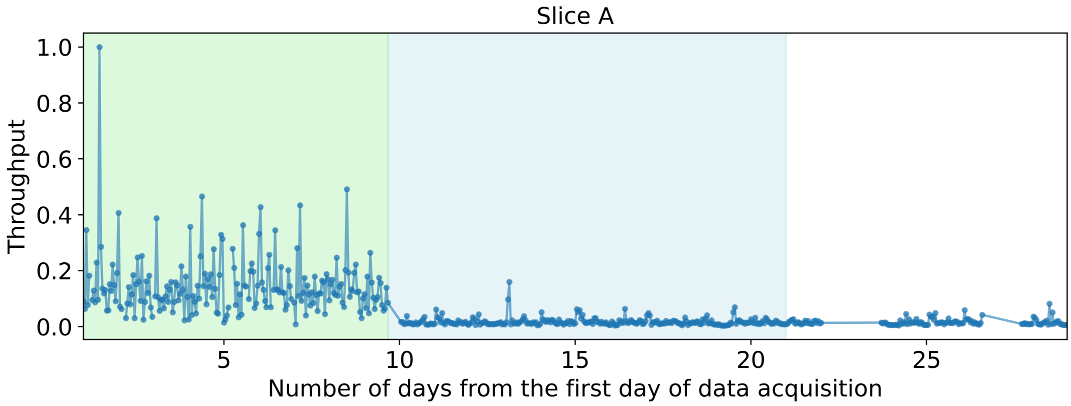


Figure 3.2: *Example trajectory of normalized throughput for Slice A.*

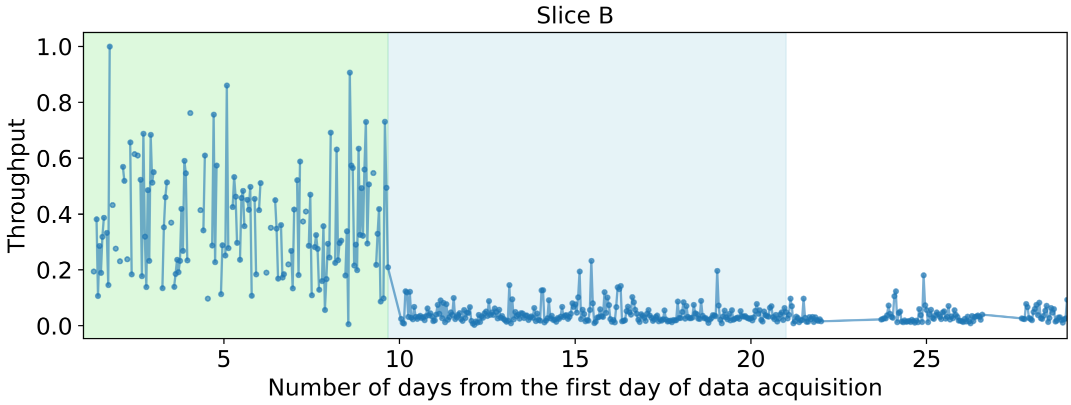


Figure 3.3: *Example trajectory of normalized throughput for Slice B.*

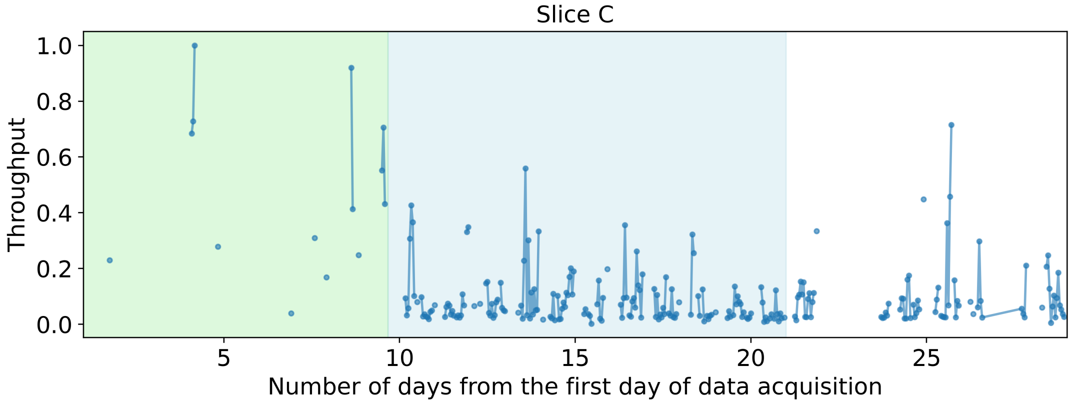


Figure 3.4: *Example trajectory of normalized throughput for Slice C.*

The configuration of a 5G BTS involves thousands of parameters. To comprehend the capabilities at the cell level, particularly concerning available resources, the most informative parameters are band and bandwidth. The band specifies the range of available

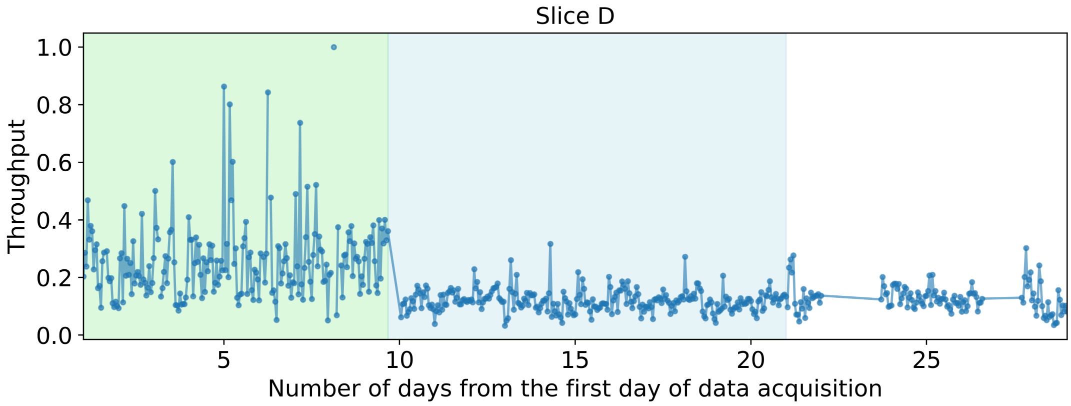


Figure 3.5: *Example trajectory of normalized throughput for Slice D.*

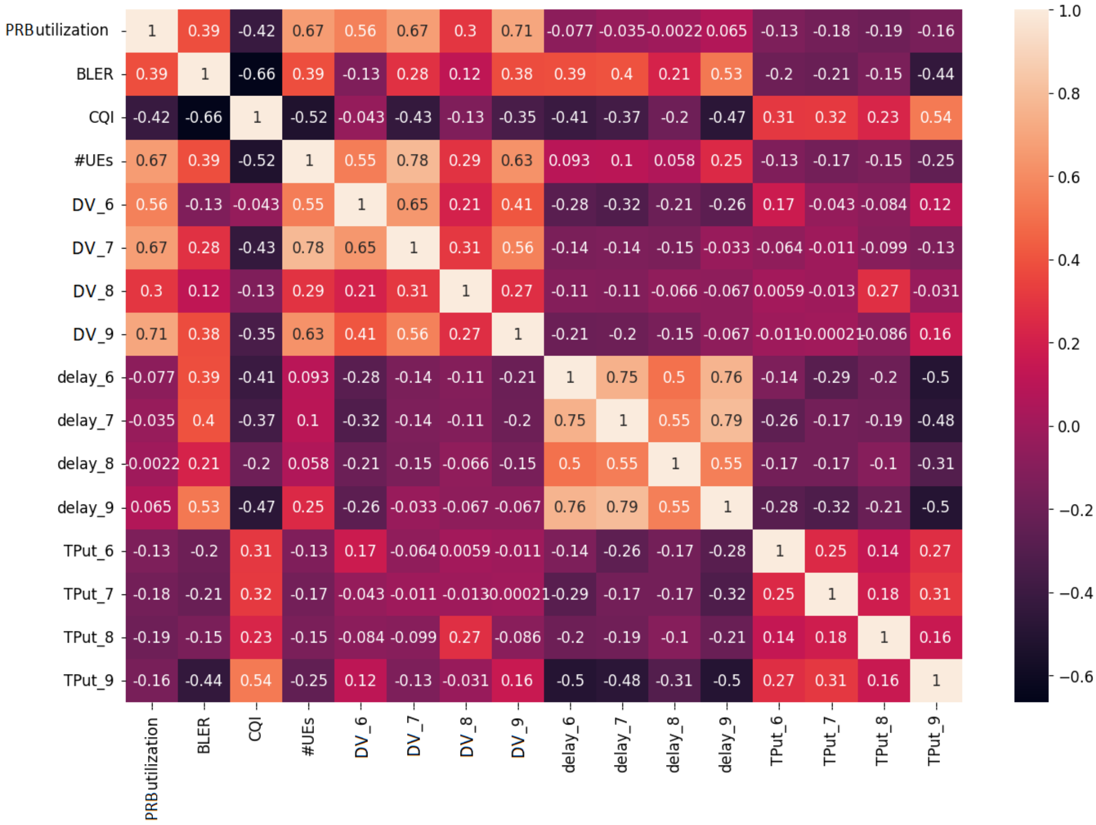
frequencies that significantly affects the propagation of radio signals, influencing its quality (e.g., CQI, BLER) and coverage. The channel frequency bandwidth refers to the width of frequency space allocated for a specific communication channel, e.g. per cell, determining the possible throughput for each user (a wider channel bandwidth allows for more data to be transmitted simultaneously). The combinations of band and bandwidth deployed in the analyzed network cluster are shown in Tab. 3.2.

Table 3.2: *Frequency bands and channel bandwidth settings in the dataset.*

	Frequency band [Mhz]	**Channel bandwidth [Mhz]**
BAND-1	2500(B41)	80, 100
BAND-2	600(B71)	15, 20

3.4 The Dependencies of Features

The examination of the dependencies between the characteristics was conducted by utilizing the Spearman correlation matrices [75] as well as the Pearson correlation coefficient displayed in Fig. 3.6 [76]. In order to maintain that the data accurately represent the situation and that the interdependencies among variables remain relatively stable despite configuration alterations, the dataset was divided based on before and after configuration modification. This study uncovers numerous linear associations between variable pairs, particularly highlighting pronounced correlations for certain pairs of delays and throughputs in different network slices. Thus, it is essential to develop an expansive model that includes the interrelationships among all the slices.



	PRButilization	BLER	CQI	#UEs	DV_6	DV_7	DV_8	DV_9	delay_6	delay_7	delay_8	delay_9	TPut_6	TPut_7	TPut_8	TPut_9
PRButilization	1	0.39	-0.42	0.67	0.56	0.67	0.3	0.71	-0.077	-0.035	-0.0022	0.065	-0.13	-0.18	-0.19	-0.16
BLER	0.39	1	-0.66	0.39	-0.13	0.28	0.12	0.38	0.39	0.4	0.21	0.53	-0.2	-0.21	-0.15	-0.44
CQI	-0.42	-0.66	1	-0.52	-0.043	-0.43	-0.13	-0.35	-0.41	-0.37	-0.2	-0.47	0.31	0.32	0.23	0.54
#UEs	0.67	0.39	-0.52	1	0.55	0.78	0.29	0.63	0.093	0.1	0.058	0.25	-0.13	-0.17	-0.15	-0.25
DV_6	0.56	-0.13	-0.043	0.55	1	0.65	0.21	0.41	-0.28	-0.32	-0.21	-0.26	0.17	-0.043	-0.084	0.12
DV_7	0.67	0.28	-0.43	0.78	0.65	1	0.31	0.56	-0.14	-0.14	-0.15	-0.033	-0.064	-0.011	-0.099	-0.13
DV_8	0.3	0.12	-0.13	0.29	0.21	0.31	1	0.27	-0.11	-0.11	-0.066	-0.067	0.0059	-0.013	0.27	-0.031
DV_9	0.71	0.38	-0.35	0.63	0.41	0.56	0.27	1	-0.21	-0.2	-0.15	-0.067	-0.011	-0.00021	-0.086	0.16
delay_6	-0.077	0.39	-0.41	0.093	-0.28	-0.14	-0.11	-0.21	1	0.75	0.5	0.76	-0.14	-0.29	-0.2	-0.5
delay_7	-0.035	0.4	-0.37	0.1	-0.32	-0.14	-0.11	-0.2	0.75	1	0.55	0.79	-0.26	-0.17	-0.19	-0.48
delay_8	-0.0022	0.21	-0.2	0.058	-0.21	-0.15	-0.066	-0.15	0.5	0.55	1	0.55	-0.17	-0.17	-0.1	-0.31
delay_9	0.065	0.53	-0.47	0.25	-0.26	-0.033	-0.067	-0.067	0.76	0.79	0.55	1	-0.28	-0.32	-0.21	-0.5
TPut_6	-0.13	-0.2	0.31	-0.13	0.17	-0.064	0.0059	-0.011	-0.14	-0.26	-0.17	-0.28	1	0.25	0.14	0.27
TPut_7	-0.18	-0.21	0.32	-0.17	-0.043	-0.011	-0.013	-0.00021	-0.29	-0.17	-0.17	-0.32	0.25	1	0.18	0.31
TPut_8	-0.19	-0.15	0.23	-0.15	-0.084	-0.099	0.27	-0.086	-0.2	-0.19	-0.1	-0.21	0.14	0.18	1	0.16
TPut_9	-0.16	-0.44	0.54	-0.25	0.12	-0.13	-0.031	0.16	-0.5	-0.48	-0.31	-0.5	0.27	0.31	0.16	1

Figure 3.6: *Pearson correlation matrix of the analyzed features [77].*

3.5 Information About Weekday

Taking information about the day of the week into account facilitates the examination of variations in network traffic patterns between weekdays and weekends. This point was also highlighted by other researchers [31]. Fig. 3.7 shows box plots that illustrate normalized throughput for weekdays and weekends, which is calculated by aggregating data from all BTS and cells in the dataset.

The variation between days becomes clear when looking at the data for each individual cell as depicted in Fig. 3.8. Despite the fact that all the traffic of the slices is routed through a single cell, each slice has distinct characteristics, indicating that they cater to different services.

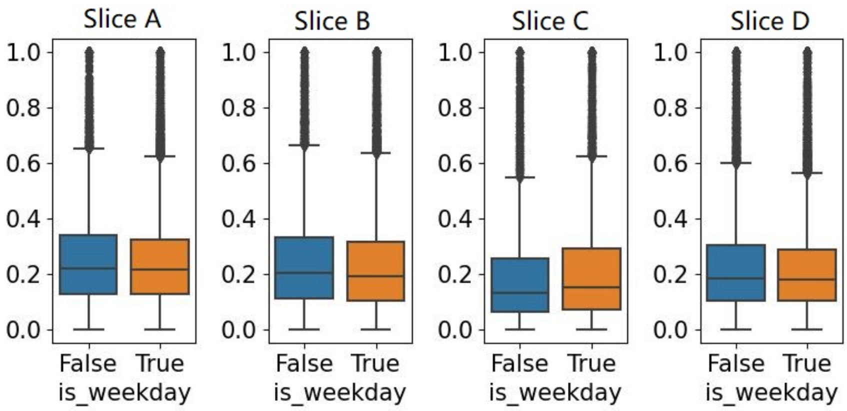


Figure 3.7: *Boxplots showing the normalized throughput for each network slice and per weekday and weekend (all cells) [77].*

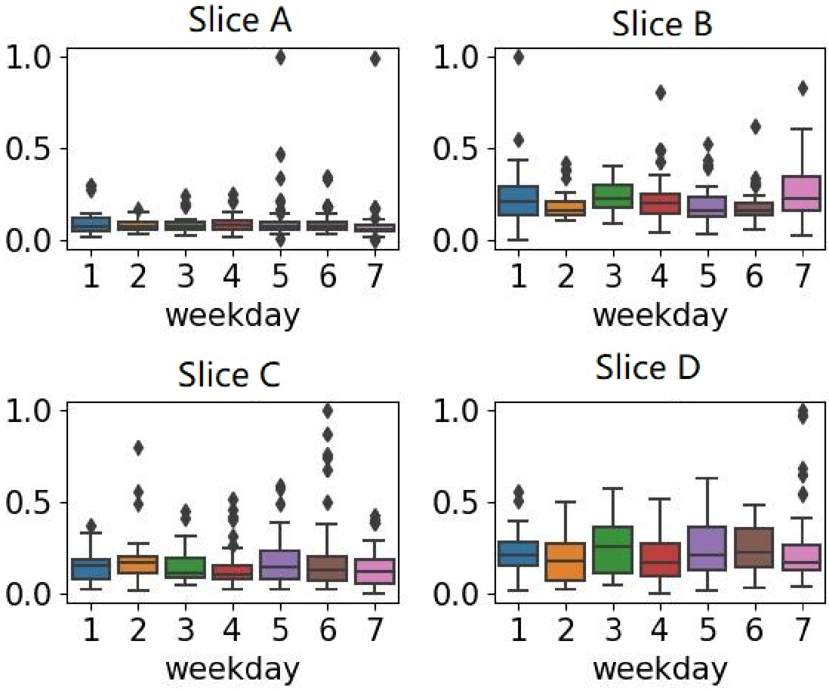


Figure 3.8: *Boxplots showing the normalized throughput by network slice and per each day of the week - 1 stands for Monday (BTS-24, CELL-1) [77].*

Furthermore, weekly patterns may vary between different cells as illustrated in Figs. 3.9 - 3.12. As an example, for Slice D (Fig. 3.12) there is a weekly cycle evident for individual days of the week. Specifically, the observations on the 4th and 11th correspond to Sundays, showing higher throughput values across whole week only for CELL-1.

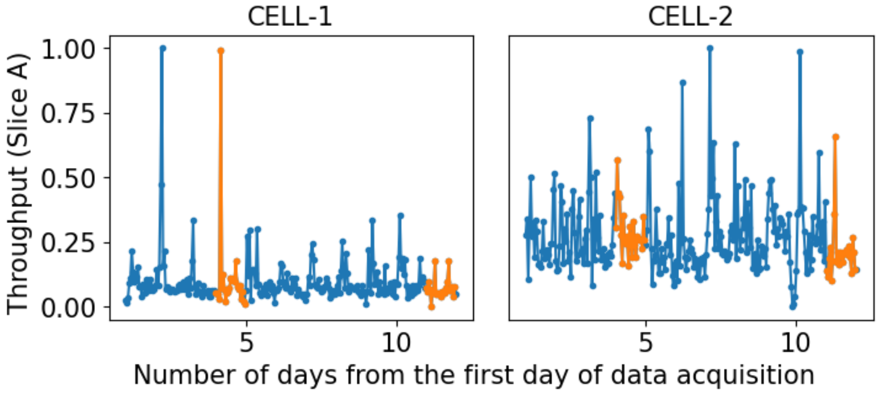


Figure 3.9: *Normalized throughput for Slice A in two specific cells (CELL-1, CELL-2). Orange points mark Sundays.*

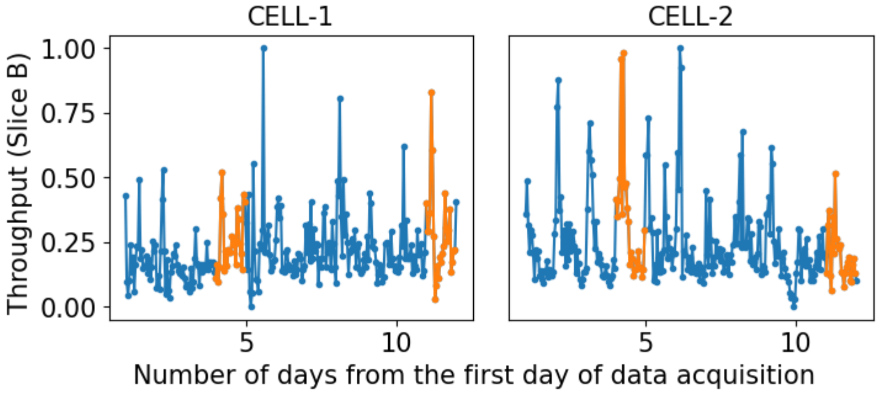


Figure 3.10: *Normalized throughput for Slice B in two specific cells (CELL-1, CELL-2). Orange points mark Sundays.*

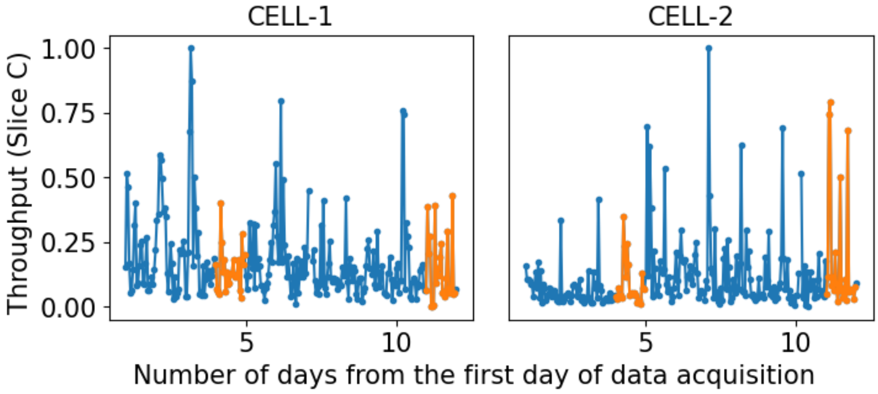


Figure 3.11: *Normalized throughput for Slice C in two specific cells (CELL-1, CELL-2). Orange points mark Sundays.*

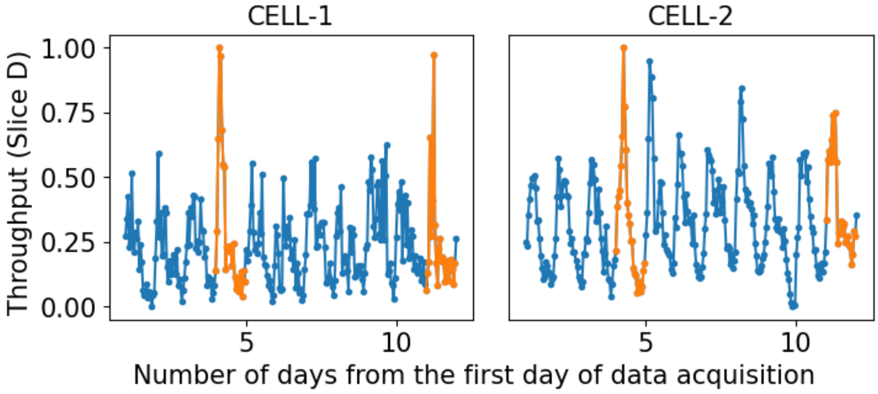


Figure 3.12: *Normalized throughput for Slice D in two specific cells (CELL-1, CELL-2). Orange points mark Sundays [77].*

The conclusion from the impact of daily seasonality on throughput and delay is that daily seasonality and unique attributes per cell must be considered when choosing a forecasting model.

3.6 Selection of Busy Hour

When making forecasts that span several months, it is crucial to take special care during the selection and preparation of the dataset. To predict peak future capacity requirements, the analysis of delay and throughput has been limited to the busiest hour of each day (Busy Hour (BH)), for the following reasons:

- CSPs require dimensioning results for the periods of highest network activity - BH, as network capacity must handle these loads,
- hourly patterns within a day may introduce unnecessary details that are not pertinent to long-term capacity planning,
- the computational complexity for training and selecting model hyperparameters is too high for data before aggregation.

Thus, the BH for each day has been identified, excluding data from less busy hours for model training and testing. The BH was determined based on the highest PRB utilization per day and cell, which indicates the peak load at the radio interface.

Furthermore, additional data selection was performed on cells with the highest load to understand their impact on the accuracy of the forecast. High-loaded cells (HL) are characterized by high PRB utilization, specifically the 0.95 quantile with PRB utilization exceeding 80%. These cells have been examined separately as they reflect BTS behavior during peak times.

Chapter 4

Methodology

This chapter elaborates on the methodologies employed in this doctoral dissertation to forecast slice-level throughput and delay as a data-driven approach to dimension 5G BTS capacity with the consideration of QoS. The goal is to identify a model that is both accurate and computationally efficient and performs well for both short-term and long-term forecasts. Using a multivariate approach, cell-specific radio and traffic conditions can be integrated to provide precise forecasts for each cell, representing the highest level of configurational granularity. However, this approach requires preprocessing multiple inputs and forecasting of exogenous variables. Consequently, there is an additional requirement to choose a one-dimensional model.

Therefore, research for this problem resolution has been split into two phases:

1. selection of one-dimensional models for exogenous variables forecasting: #UEs, QCI,
2. selection of multidimensional models for throughput and delay forecasting.

4.1 One-Dimensional Models

The primary objective of multivariate forecasting can be achieved employing multidimensional models. However, these models require exogenous variables as input. Hence, it is necessary to investigate one-dimensional models. As outlined in Sec. 3.5, telecommunications network traffic exhibits seasonal patterns (hourly and weekly seasonality), which must be taken into account in the modeling.

The method frequently employed by researchers (Sec. 2.2) is ARIMA, a technique effective for stationary signals or decomposed signals. There is also an enhanced version incorporating signal seasonality known as Seasonal AutoRegressive Integrated Moving Average (SARIMA).

Another approach worth exploring is the modular regression model known as Prophet. The reason for choosing Prophet is its ability to account for seasonality [78]. This model performs effectively for signals that show a consistent trend and simple seasonality. Moreover, it can automatically find a good set of hyperparameters for the model for efficient forecasting contrary to ARIMA, which requires manual tuning of the hyperparameters.

The most recent research method involves the latest FTSM, which are asserted to be more precise than statistical models due to being free of assumptions (no presuppositions about structures and forms). Among the many models available in the literature [53–55], Lag-Llama has been chosen.

4.1.1 SARIMA

SARIMA, or Seasonal Autoregressive Integrated Moving Average, is a flexible and popular model for time series forecasting [79, 80]. An enhancement of the non-seasonal ARIMA model, it is tailored for datasets with seasonal trends. SARIMA adeptly captures both short-term and long-term dependencies in data, which makes it a strong forecasting tool. It merges the principles of autoregressive (AR), integrated (I), and moving average (MA) models, incorporating seasonal elements.

The SARIMA model is represented as:

$$(1)\Phi(B^S)\phi(B)(x_t-\mu)=\Theta(B^S)\theta(B)\omega_t \quad (4.1)$$

The non-seasonal components are:

- Autoregressive (AR) component represented by

  $$\phi(B)=1-\phi_1B-...-\phi_pB^p$$

  describes the relationship between the current observation and a certain number of lagged observations (previous values in the time series).

- Moving Average (MA) component represented by

  $$\theta(B)=1+\theta_1B+...+\theta_qB^q$$

  describes the relationship between the current observation and the residual errors of a moving average model applied to lagged observations.

The seasonal components are:

- Seasonal Autoregressive (SAR) component represented by

$$\Phi(B^S) = 1 - \Phi_1 B^S - ... - \Phi_P B^{PS}$$

  describes the relationship between the current observation and a certain number of lagged observations at seasonal intervals.

- Seasonal Moving Average (SMA) component represented by

$$\Theta(B^S) = 1 + \Theta_1 B^S + ... + \Theta_Q B^{QS}$$

  describes the relationship between the current observation and the residual errors from a moving average model applied to lagged observations at seasonal intervals.

4.1.2 Prophet

The Prophet forecasting approach excels particularly with datasets that display clear seasonal trends and efficiently handles missing data and anomalies in its use [78]. It integrates holiday impacts and various seasonalities (annually, weekly, and daily). Consequently, non-linear trends are addressed by incorporating these factors.

Formulated as a decomposable time series model, the Prophet model is represented as:

$$X_t = g(t) + s(t) + h(t) + Z_t, \quad (4.2)$$

where

- $g(t)$ denotes the trend,
- $s(t)$ signifies periodic variations,
- $h(t)$ indicates holiday impacts,
- Z_t captures stochastic behavior.

The model facilitates the incorporation of seasonality in two ways: either additively or multiplicatively, with the latter requiring a transformation of the data using logarithms. In the simplest setup, time is usually the only regressor. Nonetheless, one can include additional regressors as long as their future values are available. In practical terms, forecasts are individually made for each regressor's future values before integrating them

Table 4.1: *Evaluated hyperparameters for Lag-Llama band level forecasting.*

Hyperparameter name	**Hyperparameter values**
Context length	32, 64, 128
Input size	1 (default)
Max context length	2048 (default)
Number of layers	1 (default)
Number of emb. per head.	32 (default)
Number head.	4 (default)
Scaling	mean (default)
Distribution output	Student's t - distribution (default)
Time features	False (default)
Dropout	None (default)
Aug. prob.	0.1 (default)
Lags sequence	["Q", "M", "W", "D", "H", "T", "S"] (default)

into a multivariate model. For optimizing the model fitting, the Prophet model employs the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm [81].

4.1.3 FTSM

For FTSM modeling two approaches have been verified:

- zero-shot prediction - an initial offline training session on synthetic data enables the model to perform predictions on fresh time series data without the need for retraining or adjusting the architecture's weights.
- fine-tuned prediction - a typical forecasting scenario in which a model is trained on numerous data points from a time series and then tested on a future segment of that same series.

Following a detailed literature review, a set of hyperparameters was chosen for zero-shot and post-tuning predictions using Lag-Llama. The band-level forecasts were performed exclusively with the zero-shot approach (Tab. 4.1), while cell-level forecasting was carried out using both methodologies (Tab. 4.2). Moreover, the results of one-shot predictions for Lag-Llama were evaluated with context lengths of 64 and 128. Employing longer context lengths is not recommended due to the usual length of the stable period.

The procedure for Lag-Llama model tuning for short-term forecasting is as follows:

1. set hyperparameters (Tab. 4.2)
2. take all observations without the last 24h for each cell for tuning the model
3. make predictions for the last 24h for each cell (test set)

Table 4.2: *Evaluated hyperparameters for Lag-Llama cell level forecasting (when there are more values than two, bolded values were taken for model tuning).*

Hyperparameter name	**Hyperparameter values**
Context length	60
Input size	1 (default)
Max context length	2048 (default)
Number of layers	8
Number of emb. per head.	**16**, 32 (default)
Number_head'	4 (default), **9**
Scaling	mean (default), **robust**
Distribution output	Student's t - distribution
Number of parallel samples	100
Time features	False (default), **True**
Dropout	None (default)
Aug. prob.	**0**, 0.1 (default)
Learning rate	0.001 (default), **0.0005**
Batch size	32
Number of the parallel samples	100
Max number of epochs	50 (default)
Shuffle buffer length	1000
Lags sequence	[0, 7, 8, 10, 11, 12, 13, 14, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 34, 35, 36, 46, 47, 48, 50, 51, 52, 55, 57, 58, 59, 60, 61, 70, 71, 72, 83, 94, 95, 96, 102, 103, 104, 117, 118, 119, 120, 121, 142, 143, 144, 154, 155, 156, 166, 167, 168, 177, 178, 179, 180, 181, 334, 335, 336, 362, 363, 364, 502, 503, 504, 670, 671, 672, 718, 719, 720, 726, 727, 728, 1090, 1091, 1092]

4.2 Multidimensional Models

Various multivariate predictive models have been validated to forecast throughput and delay. According to the latest literature in the domain described in Sec. 2.2, the main options have been narrowed down to multivariate Autoregressive Moving Average (ARMA) models and neural networks.

Vector Autoregressive Moving Average with eXogenous regressors model (VARMAX) has been selected because of following advantages:

- ability to include multiple input variables,
- ability to forecast multiple variables,
- ability to understand the temporal relationship between variables,
- is an extension to well known one-dimensional models (ARMA),
- requires less data than neural networks.

There are also some assumptions that create problems during implementation:

- requires complete and weakly stationary data that creates a need for data preprocessing (decomposition), which can lead to information loss,
- requires complete and evenly time distributed data, which creates a need of data selection or gap filling.

Neural networks (LSTM) have been selected because of the following advantages:

- ability to include multiple input variables,
- ability to forecast multiple variables,
- based on recurrencial structures and considers seasonality,
- prevents the vanishing gradient problem.

4.2.1 VARMAX

A time series $\{\boldsymbol{X_t}\}$ is a m-variate ARMA($p,\ q$) process (called also vector ARMA, Vector Autoregressive Moving-Average (VARMA)) formulated in the following way [82]:

$$\Phi(B)\boldsymbol{X_t} = \Theta(B)\boldsymbol{Z_t}, \tag{4.1}$$

where $\{X_t\}$ is a stationary solution of difference equations (4.1), where

- $\Phi(z) := I - \Phi_1 z - ... - \Phi_p z^p$, where $\Phi_1, ..., \Phi_p$ are $m \times m$ matrices,
- $\Theta(z) := I - \Theta_1 z - ... - \Theta_q z^q$, where $\Theta_1, ..., \Theta_q$ are $m \times m$ matrices.

Moreover:

- I is $m \times m$ identity matrix,
- B is the backward shift operator,
- $\{\boldsymbol{Z_t}\}$ is multivariate white noise sequence.

Prior to employing the VARMAX model, it is essential to decompose the variables to ensure that the data align with the model requirements. The complete procedure flow chart used to decompose the data and create the VARMAX model is shown in Fig. 4.1.

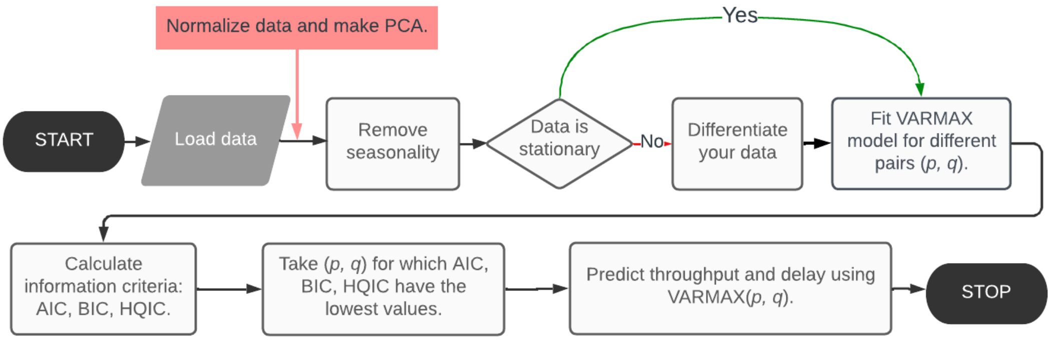

Normalize data and make PCA.
START
Load data
Remove seasonality
Data is stationary
Yes
No
Differentiate your data
Fit VARMAX model for different pairs (*p*, *q*).
Calculate information criteria: AIC, BIC, HQIC.
Take (*p*, *q*) for which AIC, BIC, HQIC have the lowest values.
Predict throughput and delay using VARMAX(*p*, *q*).
STOP

Figure 4.1: *Diagram of the method employing the VARMAX model.*

The first step after loading the data ("Normalize data and make PCA" - red box) is not necessary from a modeling point of view, however, it brings some benefits. Overall, Principal Component Analysis (PCA) serves as a method for dimensionality reduction [83], aiming to determine whether the model's number of parameters can be minimized. During the research (results provided in Chapter 6) it was verified that it is beneficial because reducing the number of variables makes the estimation of parameters of VARMAX faster. Moreover, in this case, the quality of the prediction is not significantly different. In addition, PCA allows to counteract the so-called curse of dimensionality.

The VARMA model requires that the inputs are stationary, which can be verified by performing the Dickey-Fuller augmented test [84]. As the network data are seasonal, the necessary step is to remove the seasonality component and verify afterwards.

The procedure based on Maximum Likelihood Estimation (MLE) is utilized to determine the coefficients of the model for the appropriate orders (p, q). Various combinations of (p, q) are assessed and the one that minimizes the information criteria is chosen. The information criteria taken into account include the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and the Hannan-Quinn Information Criterion (HQIC) [85]. Subsequently, the VARMAX(p, q) model is applied for the selected values of p and q.

4.2.2 LSTM

This section focuses on evaluating various neural network structures with different layers and units. Previous research indicates that there is no universally accepted method for determining the number of training epochs [86]. Thus, following preliminary tests, the maximum number of training epochs has been set to 1000, with a provision to halt training if loss does not decrease over 20 consecutive epochs. Prior to training the neural networks, the data are normalized and subjected to seasonal decomposition, a process similar to that used in VARMAX (Sec. 4.2.1).

RNN are commonly applied in time series prediction tasks [27, 31, 87–89]. However, when dealing with long sequences, there is an issue known as the gradient-vanishing problem [27, 87, 90, 91], which arises from the small values of partial derivatives calculated for weights, preventing them from being updated [92]. In the context of recursive networks, a long time horizon T implies that the initial observation x_1 and the final observation x_T are distant from each other. To mitigate the vanishing gradient problem, the LSTM unit can be used [93], which represent a variation of the traditional RNN.

The neural network architectures discussed in this section are based exclusively on LSTM units and dense layers. The particular set of hyperparameters that was evaluated is detailed in Tab. 4.3. During preliminary tests, it has been determined to evaluate networks with a maximum of three layers. The choice of shallow structures is supported by the reduced training time of the model and their potential for further development in the case of inefficiency (e.g., high loss).

4.2.3 CNN-BiLSTM

The next architecture explored combines convolutional layers and Bidirectional Long Short-Term Memory (BiLSTM). In a CNN, neurons are exclusively linked to a filter,

Table 4.3: *Evaluated hyperparameters for LSTM block structures [77].*

Hyperparameter name	**Hyperparameter values**
LSTM-number of layers	1, 2, 3
LSTM-units	[50], [50, 50], [50, 50, 20]
LSTM-dropout rate	0.2
LSTM-activation	tangent
LSTM-recurrent activation	sigmoid
Learning rate	0.01, 0.001
Optimizer	SGD, Adam
Batch size	24
Loss	MAE, MSE

which represents a specific area in the preceding layer, unlike the traditional fully connected neural network [89, 94]. Similarly to the previous section, various combinations of parameters have been tested (Tab. 4.4).

Table 4.4: *Verified hyperparameters for the CNN-BiLSTM architecture.*

Hyperparameter name	**Hyperparameter value**
CONV1 (filters, kernel size)	(128, 4), (256, 4)
CONV2 (filters, kernel size)	(64, 2), (128, 4)
CONV-activation	relu, tanh
Pooling size	2
BiLSTM-number of layers	1
BiLSTM-units	100
BiLSTM-activation	relu
BiLSTM-recurrent activation	relu
Dropout rate	None, 0.1
Batch size	24
Learning rate	0.001, 0.0001
Optimizer	Adam, SGD
Loss	MSE, MAE

4.3 Research Environment and Tools

The research and development described in this doctoral dissertation were carried out using Python 3.10 with Jupyter Notebooks. A variety of publicly available libraries have been utilized (minimal library version is listed if required):

- Data normalization, regression models, statistical metrics (e.g. MAE): sklearn, xgboost,
- Data processing and vizualization: matplotlib v. 3.8.4, numpy v. 1.26.4, pandas v. 2.2.1, plotly v. 5.20.0, scikit-learn v. 1.4.1, scipy v. 1.13.0, seaborn v. 0.13.2,
- ARIMA: pmdarima v. 2.0.4,
- PROPHET: prophet,
- VARMAX, SARIMA and ACF testing: statsmodels v. 0.14.1,
- Parameter optimization: optuna v. 3.6.1,
- Neural networks: tensorflow v. 2.16.1, keras v. 3.3.3,
- FTSM predictions: gluonts, autogluon, torch v. 2.0.0, wandb, huggingfacehub,
- CUDA Toolkit v. 12.4 development environment for GPU usage.

Moreover, the hardware unit that has been used in the research included:

- 16 CPU: Intel(R) Core(TM) i7-7820X 3.60GHz,
- 64 GB RAM memory,
- 1 GPU NVIDIA GeForce GTX 1080 card with 8GB RAM memory.

Chapter 5

Environmental Variables Forecasting

The variables selected for the input to a delay and throughput forecasting model (Tab. 3.1) need to be preprocessed and in the case of exogenous variables also forecasted. To be specific, it is necessary to forecast #UEs, CQI with the use of a one-dimensional model as input for multidimensional forecasting models. These two KPIs are measured only at the cell level; therefore, this analysis is not available for slice-based granularity.

After literature research (Sec. 2.2), several models have been selected for initial studies as described in Sec. 4.1. In this chapter, an evaluation of these models is presented, and the best is selected for the overall dimensioning framework presented in this doctoral dissertation.

5.1 Selection of Models for Fine-grained Comparison

Initial model comparison has been done for data aggregated in the following way:

- all cells (that have enough data for modeling),
- high-loaded cells (Sec. 3.6).

The data is also aggregated from the cell level to a band level (Tab. 3.2) by calculating the mean value over all timeseries over time. This approach enabled evaluating each model on most of the data gathered and selecting models for further cell-level analysis.

Exemplary results per band for each model have been presented for UEs forecasting in Figs. (BAND-1) 5.1 - 5.4, (BAND-2) 5.9 - 5.12 and for CQI forecasting in Figs. (BAND-1) 5.5 - 5.8, (BAND-2) 5.13 - 5.16.

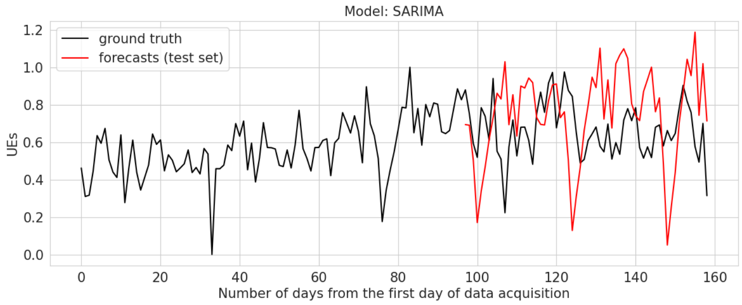


Figure 5.1: *Forecast of #UEs for Band-1 with SARIMA.*

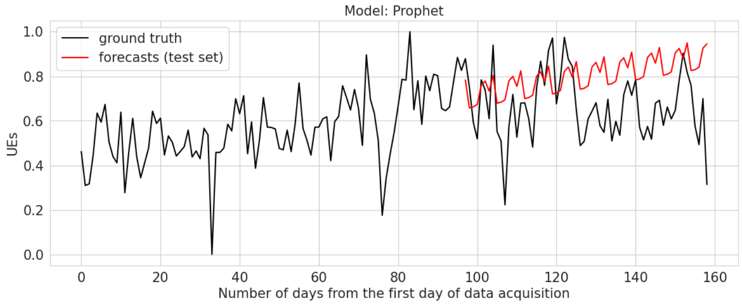


Figure 5.2: *Forecast of #UEs for Band-1 with Prophet.*

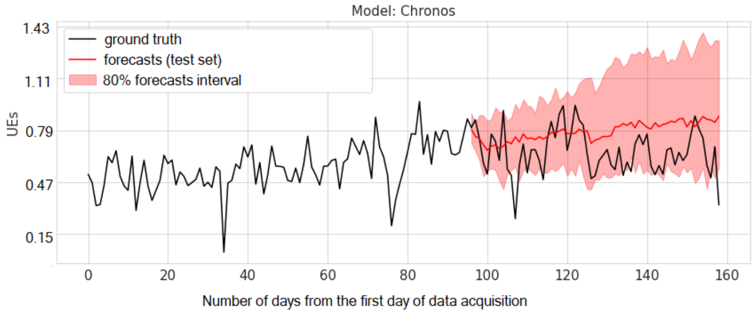


Figure 5.3: *Forecast of #UEs for Band-1 with Chronos.*

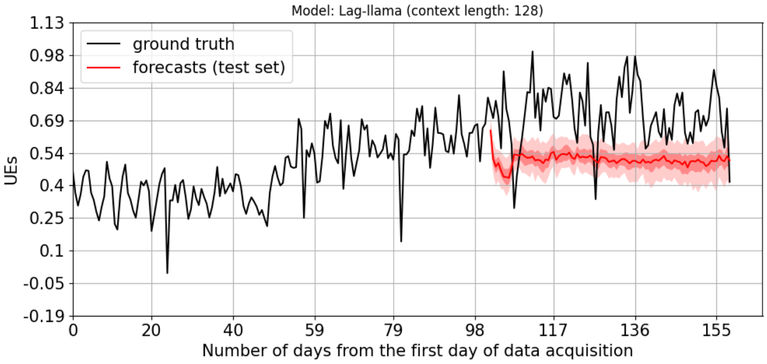


Figure 5.4: *Forecast of #UEs for Band-1 with Lag-Llama.*

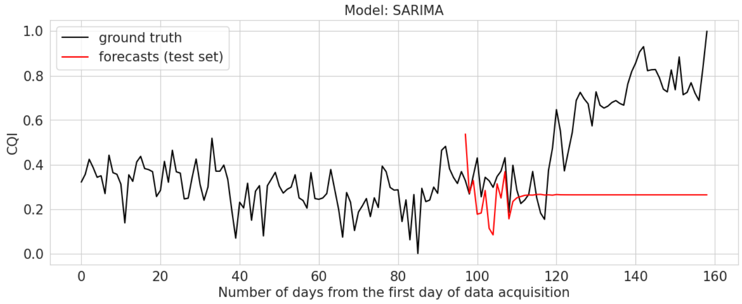


Figure 5.5: *Forecast of CQI for Band-1 with SARIMA.*

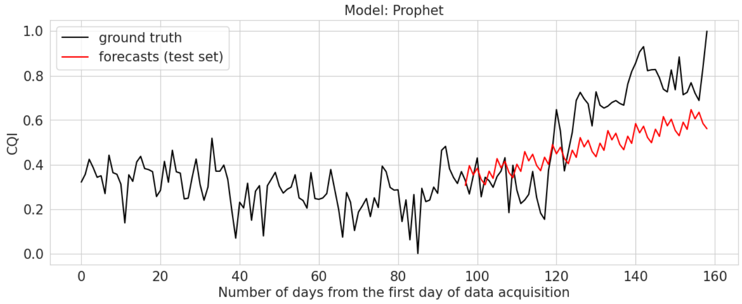


Figure 5.6: *Forecast of CQI for Band-1 with Prophet.*

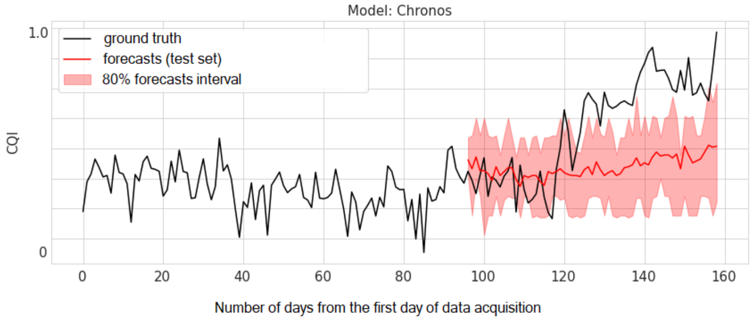


Figure 5.7: *Forecast of CQI for Band-1 with Chronos.*

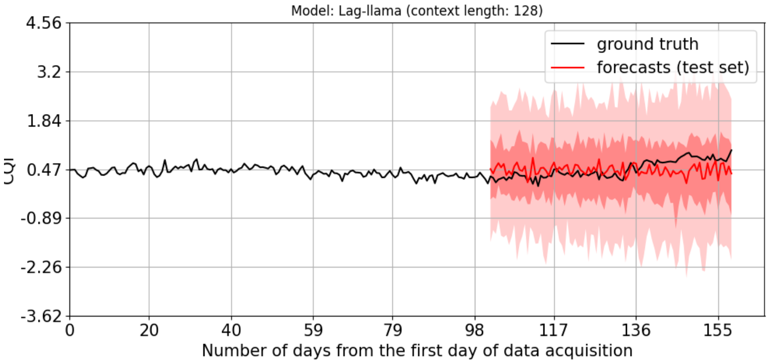


Figure 5.8: *Forecast of CQI for Band-1 with Lag-Llama.*

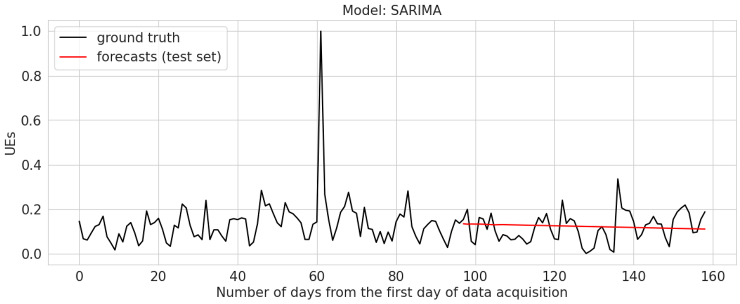


Figure 5.9: *Forecast of #UEs for Band-2 with SARIMA.*

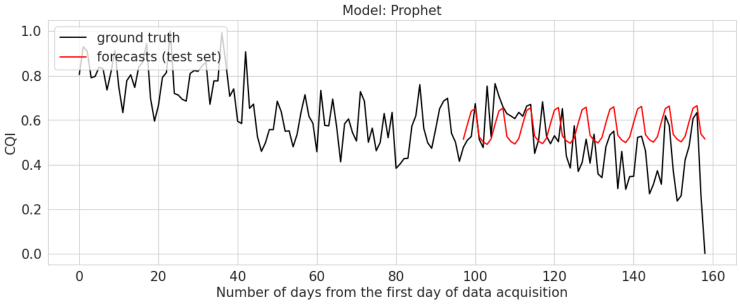


Figure 5.10: *Forecast of #UEs for Band-2 with Prophet.*

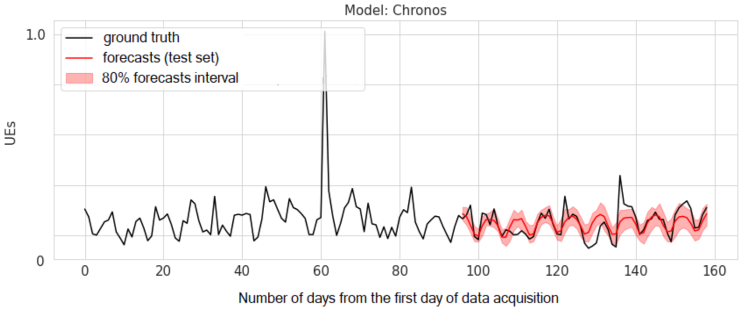


Figure 5.11: *Forecast of #UEs for Band-2 with Chronos.*

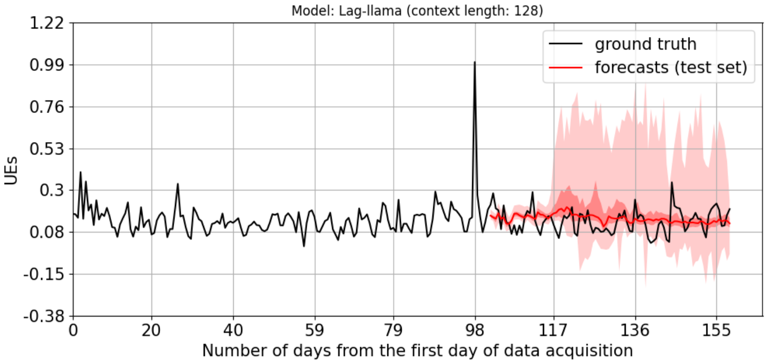


Figure 5.12: *Forecast of #UEs for Band-2 with Lag-Llama.*

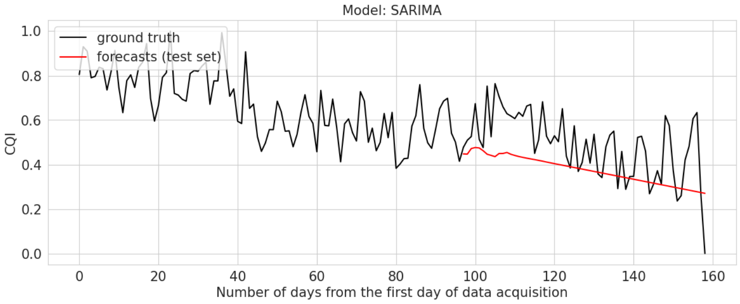


Figure 5.13: *Forecast of CQI for Band-2 with SARIMA.*

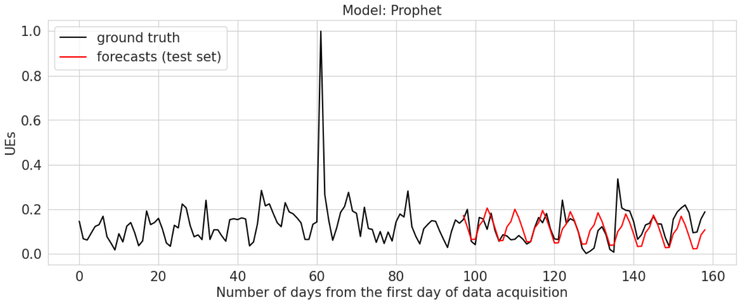


Figure 5.14: *Forecast of CQI for Band-2 with Prophet.*

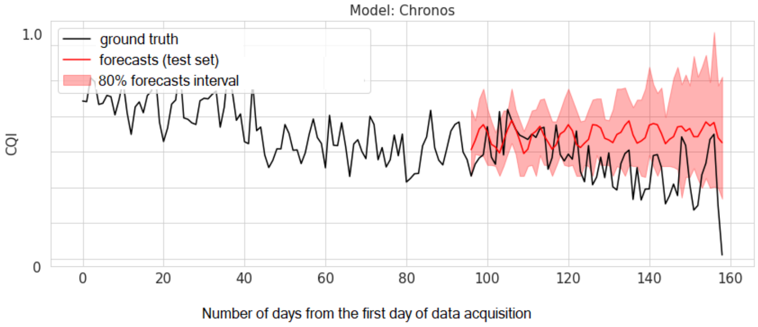


Figure 5.15: *Forecast of CQI for Band-2 with Chronos.*

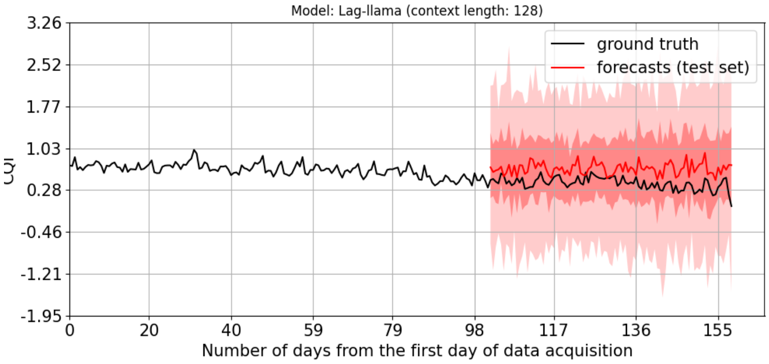


Figure 5.16: *Forecast of CQI for Band-2 with Lag-Llama.*

Analyzing the plots of the actual and forecasted values, it is clear that SARIMA produces the least accurate results because it fails to capture the seasonality of the signal, resulting in values near the average signal levels. In contrast, the other three models successfully align with seasonal patterns. To quantitatively assess the accuracy of the predictions, all models were evaluated using the normalized Mean Absolute Error (nMAE) and Mean Absolute Percentage Error (MAPE) metrics (Tabs. 5.1, 5.2).

Table 5.1: *Comparative analysis of nMAE for one-dimensional models in long-term UEs and CQI forecasting across all cells averaged by band.*

Band	KPI	SARIMA	Prophet	Chronos	Lag-Llama
BAND-1	UEs	0.252	0.163	0.181	0.256
	CQI	0.627	0.529	0.444	0.368
BAND-2	UEs	0.059	0.079	0.072	0.052
	CQI	0.213	0.356	0.295	0.250

Table 5.2: *Comparative analysis of MAPE for one-dimensional models in long-term UEs and CQI forecasting across all cells averaged by band.*

Band	KPI	SARIMA	Prophet	Chronos	Lag-Llama
BAND-1	UEs	0.166	0.154	0.172	0.173
	CQI	0.028	0.039	0.033	0.027
BAND-2	UEs	0.211	0.775	0.688	0.374
	CQI	0.047	0.058	0.048	0.041

Examining nMAE and MAPE indicates that Lag-Llama demonstrates the highest accuracy in most scenarios, making it the preferred model for fine-grained analysis. SARIMA had already been ruled out because it did not adhere to the seasonality pattern. Although Chronos yields more precise results in more cases compared to Prophet, the differences are marginal. Given that Chronos belongs to the same category of models as Lag-Llama, Prophet is chosen for further comparison to include a model of a different type.

5.2 Comparison of One-dimensional Short-term Forecast

In both cases of #UEs and CQI, it can be observed that the median normalized MAE (nMAE) for the one-shot Lag-Llama model is higher and the ranges are wider than for the Prophet model (Fig. 5.17 and Fig. 5.18). Consequently, Prophet outperforms the one-shot Lag-Llama. This conclusion is supported by the figures that depict the nMAE for various models, including a breakdown by cells (Fig. 5.19 and Fig. 5.20). However, after parameter tuning (Sec. 4.1.3), the Lag-Llama model demonstrates a slight improvement in forecasting #UEs compared to Prophet, as shown by a lower median nMAE.

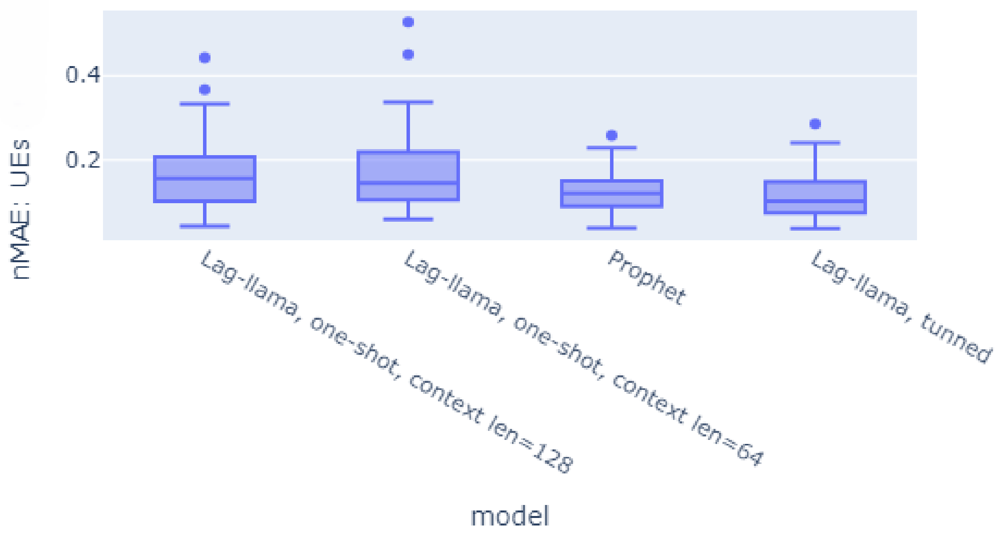


Figure 5.17: *The comparison nMAE for short-term UEs forecasting: Lag-Llama vs Prophet.*

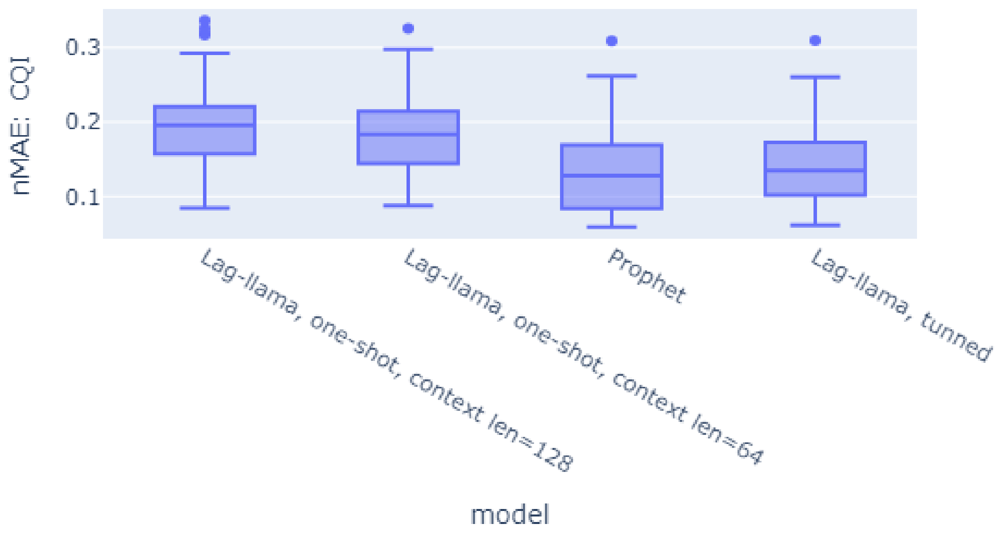


Figure 5.18: *The comparison nMAE for short-term CQI forecasting: Lag-Llama vs Prophet.*

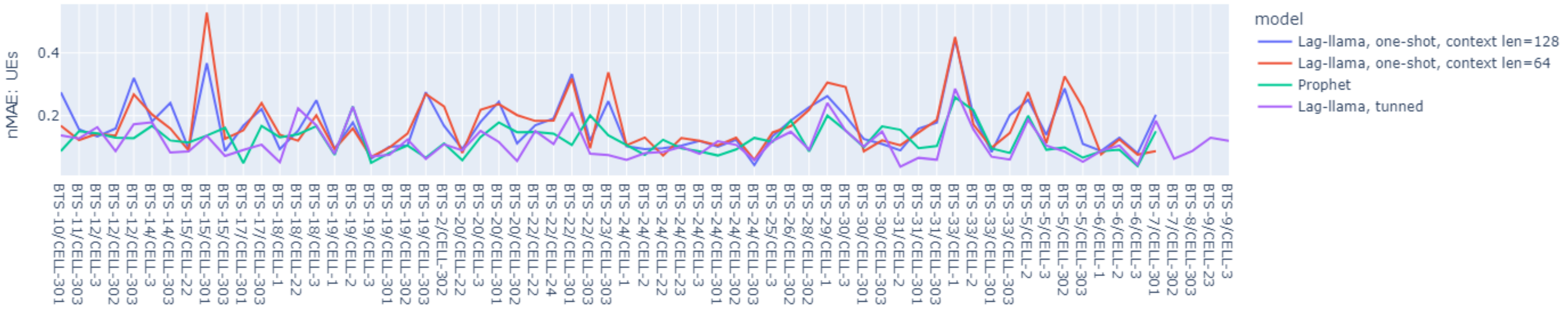


Figure 5.19: *The comparison of the Lag-Llama (one shot and tuned) and Prophet models efficiency for UEs forecasting.*

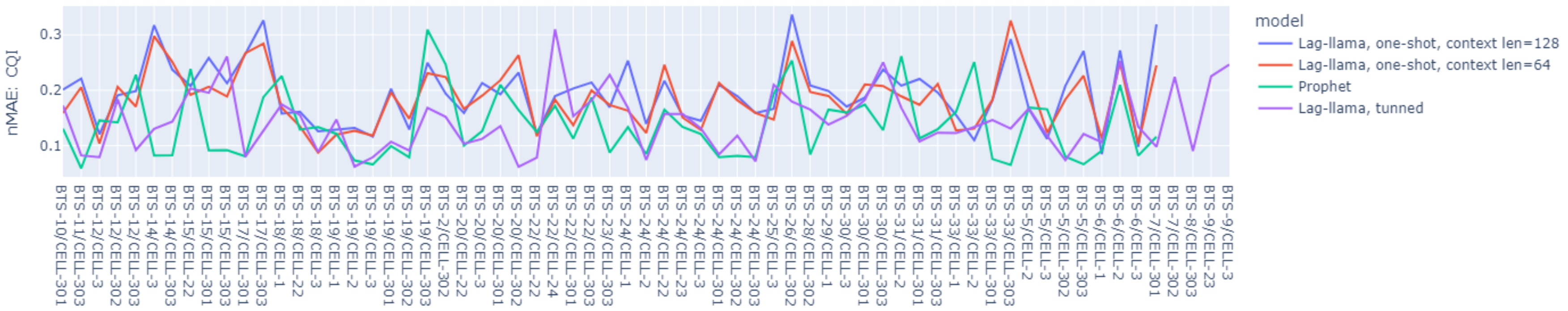


Figure 5.20: *The comparison of the Lag-Llama (one shot and tuned) and Prophet models efficiency for CQI forecasting.*

Further investigation of cell-level data shows that the evaluation metrics are similar for Prophet and Lag-Llama when there are no complicated trends and seasonality is highly visible in the data (Figs. 5.21 - 5.24).

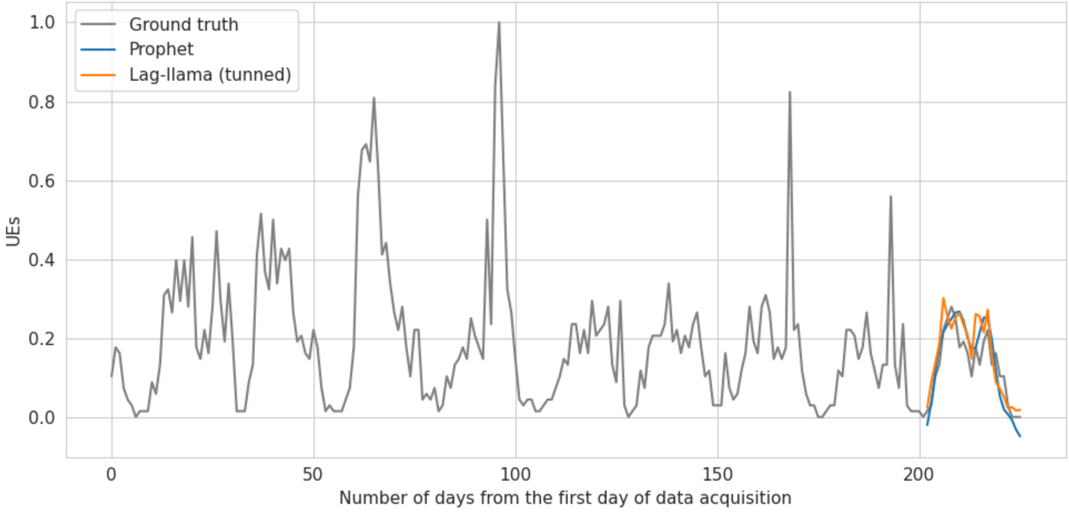


Figure 5.21: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-6, CELL-3).*

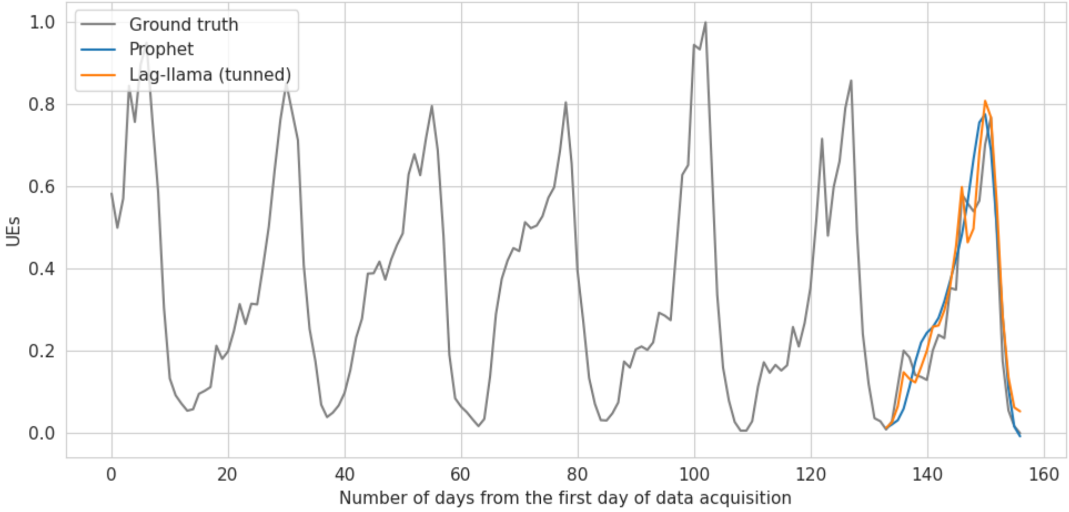


Figure 5.22: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-5, CELL-303).*

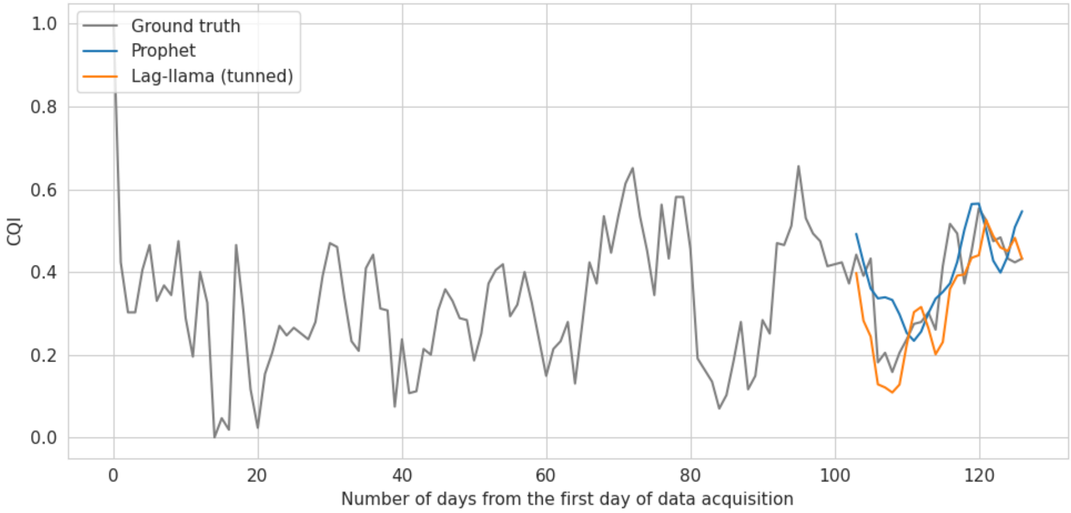


Figure 5.23: *Illustrative example of Prophet and Lag-Llama CQI forecast (BTS-19, CELL-2).*

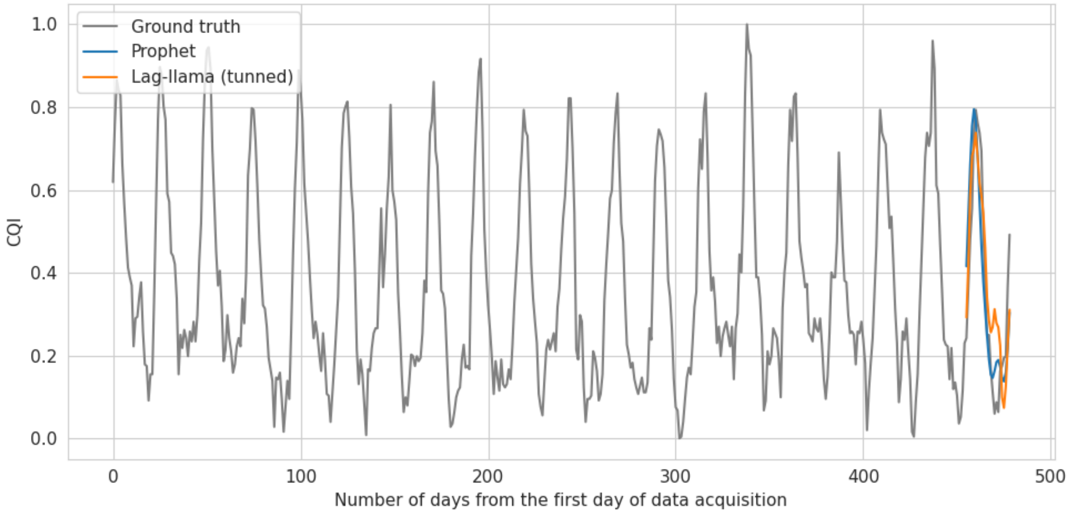


Figure 5.24: *Illustrative example of Prophet and Lag-Llama CQI forecast (BTS-19, CELL-301).*

However, there are examples with significant differences between Lag-Llama and Prophet, shown and discussed below.

5.2.1 Results where Prophet is Outperforming Lag-Llama

For forecasting the number of UEs, Prophet outperformed Lag-Llama significantly in the specific instance illustrated in Fig. 5.25 and for CQI as depicted in Fig. 5.26. The data exhibit a clear linear and seasonal trend, making the optimization problem for Prophet straightforward to solve, yielding reliable results.

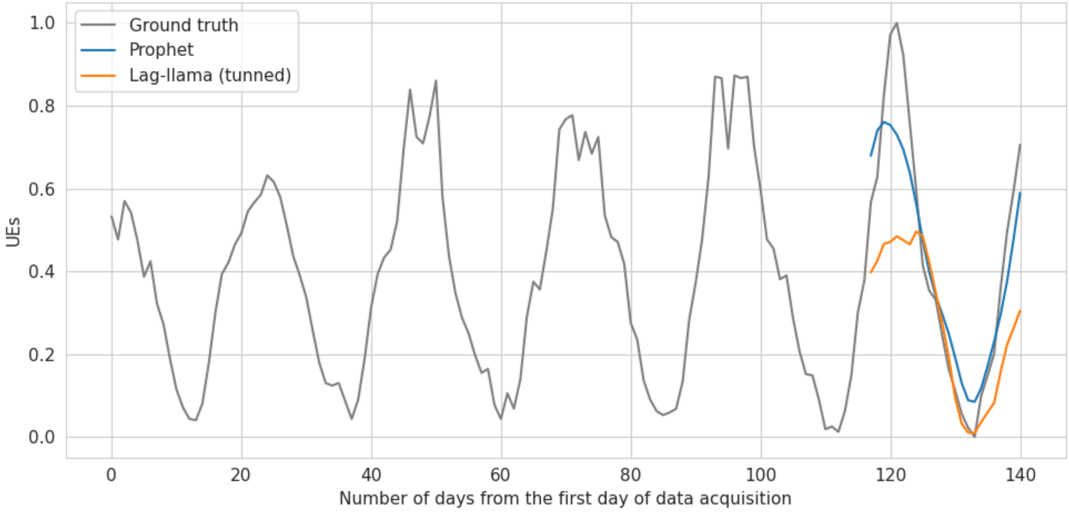


Figure 5.25: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-22, CELL-301).*

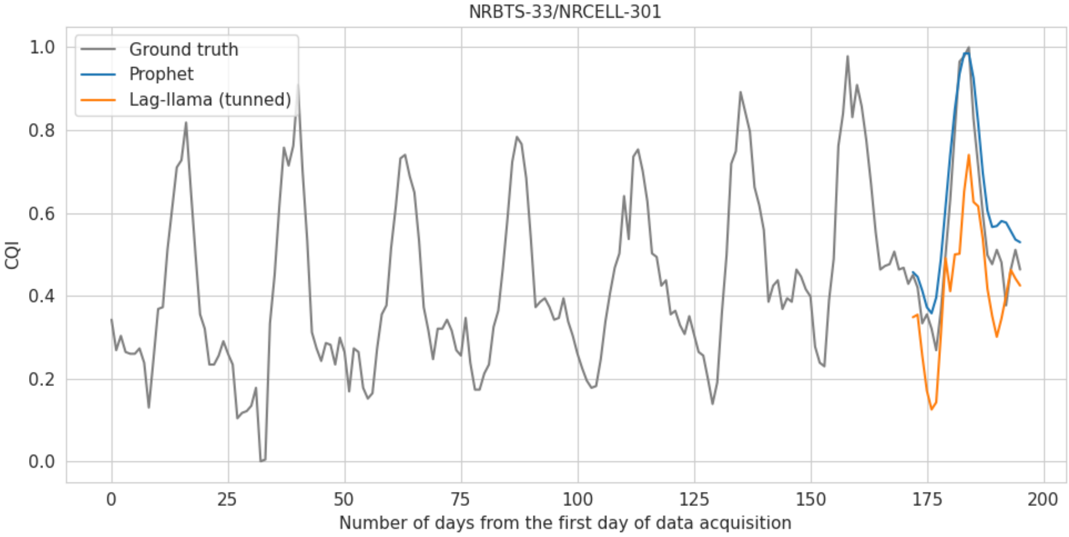


Figure 5.26: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-33, CELL-301).*

5.2.2 Results where Lag-Llama is Outperforming Prophet

In cases of unusual data characteristics, Lag-Llama outperforms Prophet, since Prophet excels with recurring trends and clear seasonality. From the observations, Prophet struggles with nonmonotonic trends. For example, for any signals with a minimum values close to zero, Prophet has a higher risk of forecasting negative values. This issue does not arise with the tuned Lag-Llama, as it is calibrated with actual data and understands the characteristics and patterns. Moreover, for any signals exhibiting a significant pattern change, Prophet may be ineffective since it attempts to fit the model to the entire trajectory, while Lag-Llama adjusts the weights to prioritize the "recent past" over the "distant past". Exemplary results for UE forecasting are illustrated in Figs. 5.27 - 5.29 and for CQI in Figs. 5.30 - 5.32.

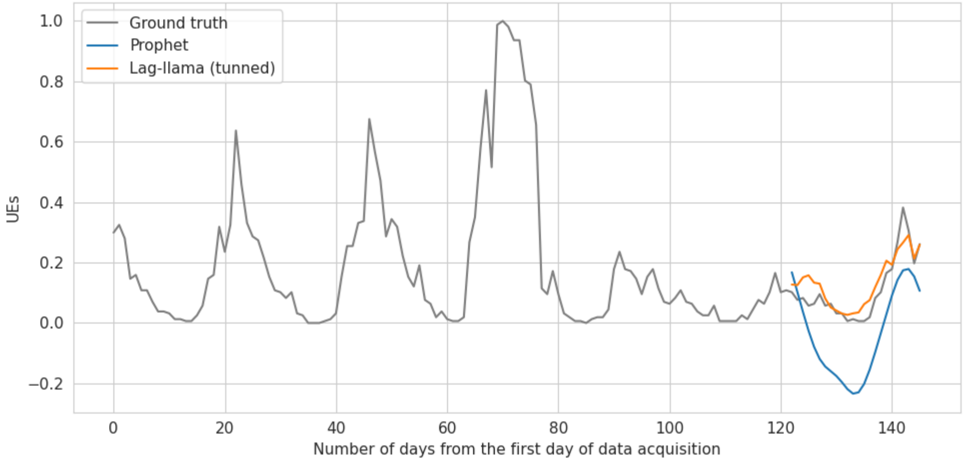


Figure 5.27: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-31, CELL-2).*

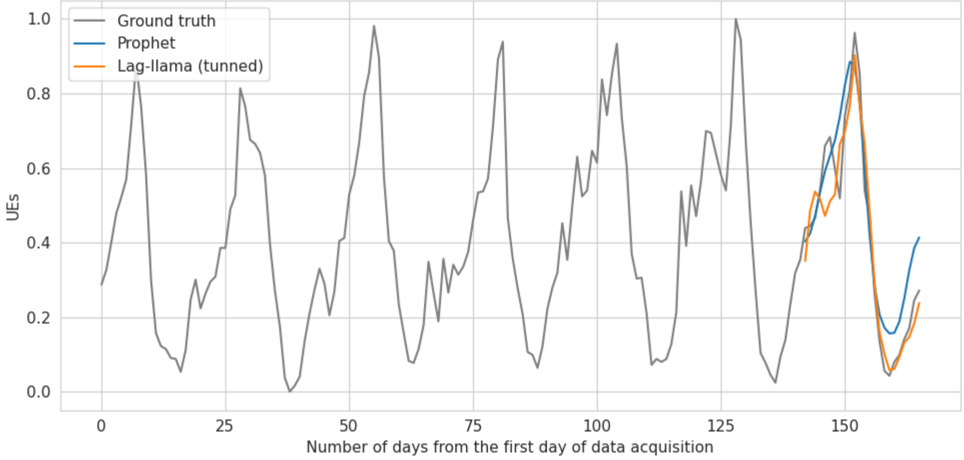


Figure 5.28: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-33, CELL-303).*

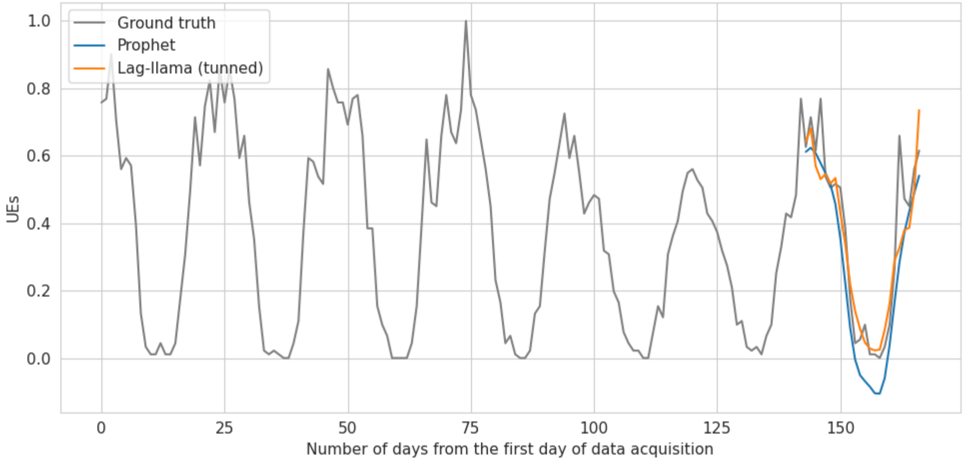


Figure 5.29: *Illustrative example of Prophet and Lag-Llama UEs forecast (BTS-31, CELL-301).*

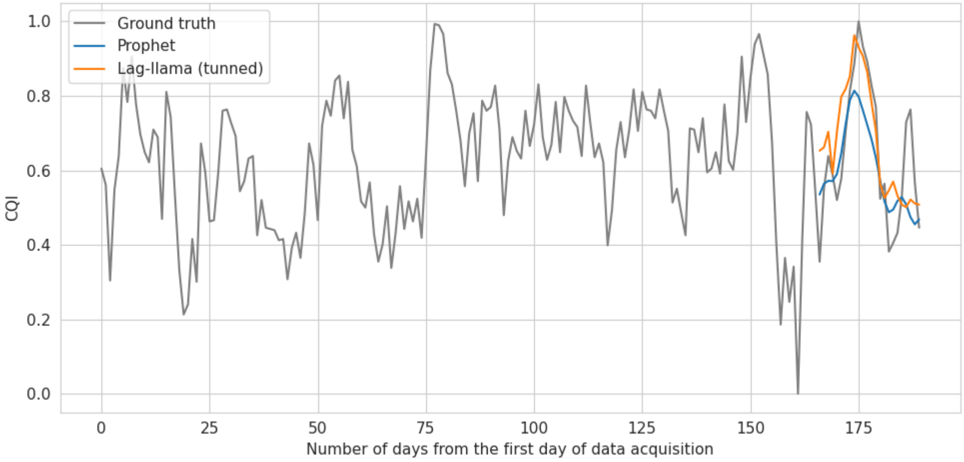


Figure 5.30: *Illustrative example of Prophet and Lag-Llama CQI forecast (BTS-20, CELL-22).*

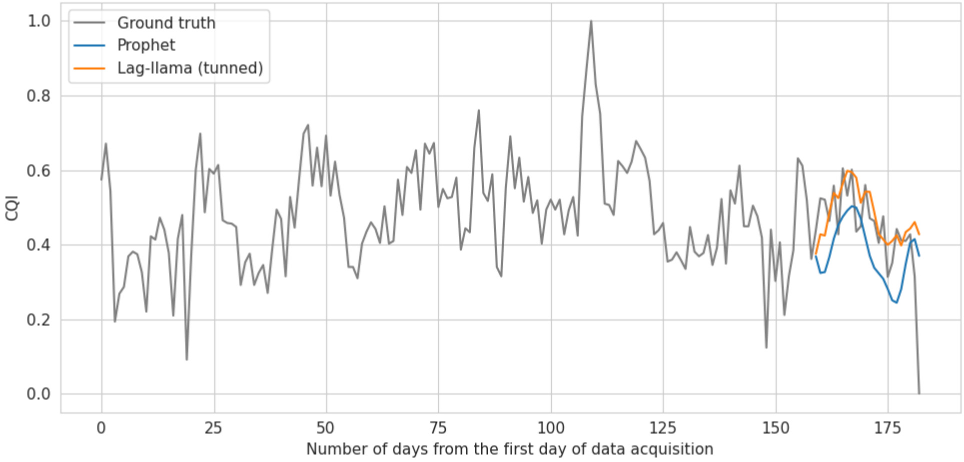


Figure 5.31: *Illustrative example of Prophet and Lag-Llama CQI forecast (BTS-22, CELL-22).*

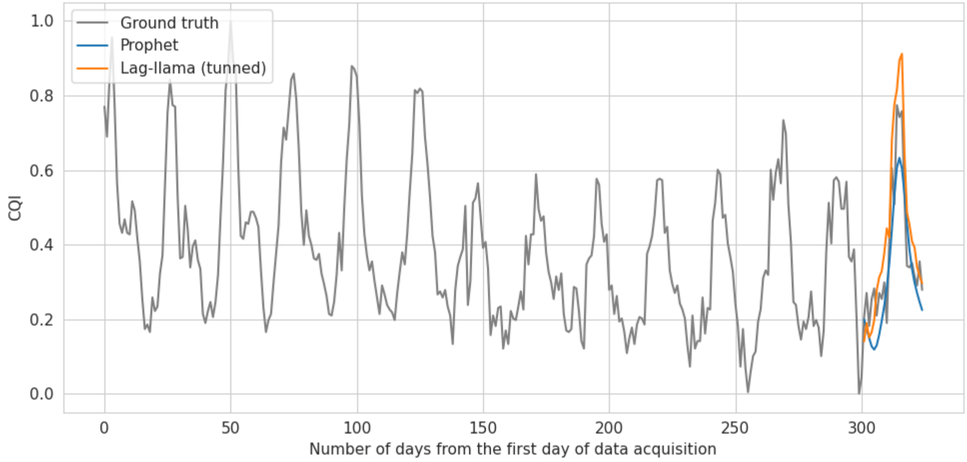


Figure 5.32: *Illustrative example of Prophet and Lag-Llama CQI forecast (BTS-19, CELL-302).*

5.3 Comparison of One-dimensional Long-term Forecast

As shown in Sec. 5.2, Lag-Llama results are more accurate for #UEs and CQI forecasts after tuning than for one shot. For that reason, one-shot forecasts for long-term cell-level modeling have been excluded from further research.

The fine tuning procedure is as follows, with two approaches for data tuning that have been tested (difference between both approaches is step 3 (a) and (b)):

1. split the data for training set (for tuning: *SET-T*) and test set (for forecasting: *SET-F*) for each band
2. set hyperparameters (analyzed ranges where given in Tab. 4.2)
3. take
   (a) 158 first observations for each cell from *SET-T* for tuning the model
   (b) all observations for each cell from *SET-T* for tuning the model. The observations from test set to exclude cases that we have the cells from the same BTS in training set and test set have been filtered.
4. make forecasts for *SET-F*.

To evaluate the models, nMAE and MAPE were calculated for each cell. The initial comparison of forecast accuracy is presented as the percentage of cells where the #UEs and CQI forecasts are more accurate with Lag-Llama compared to Prophet (nMAE in Figs. 5.33 and 5.34 and MAPE on Figs. 5.35 and 5.36).

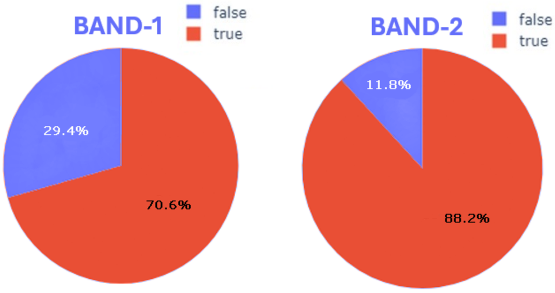


Figure 5.33: *The percentage of cells where the UEs forecasts for Lag-Llama are better than for Prophet calculated with nMAE (Lag-Llama without limited time range for model tuning).*

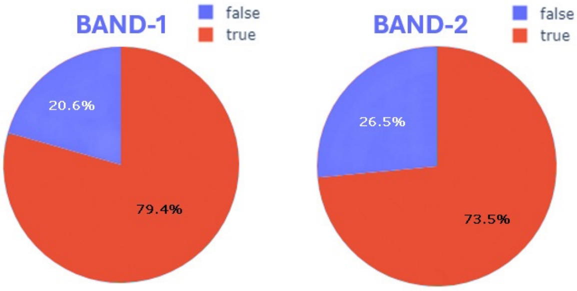


Figure 5.34: *The percentage of cells where the CQI forecasts for Lag-Llama are better than for Prophet calculated with nMAE (Lag-Llama without limited time range for model tuning).*

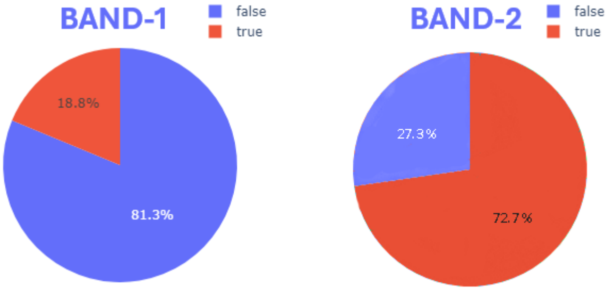


Figure 5.35: *The percentage of cells where the UEs forecasts for Lag-Llama are better than for Prophet calculated with MAPE (Lag-Llama without limited time range for model tuning).*

These results show that for most of the 5G cells the Lag-Llama gives more accurate results. Next, an in-depth analysis, with a cell-by-cell comparison, is presented in Figs. 5.37 - 5.44.

Finally, the results have been summarized on boxplots (Figs. 5.45 - 5.48).

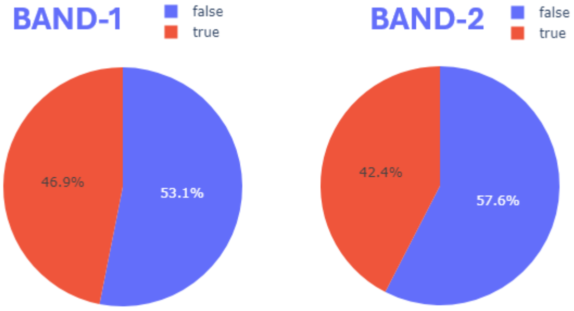


Figure 5.36: *The percentage of cells where the CQI forecasts for Lag-Llama are better than for Prophet calculated with MAPE (Lag-Llama without limited time range for model tuning).*

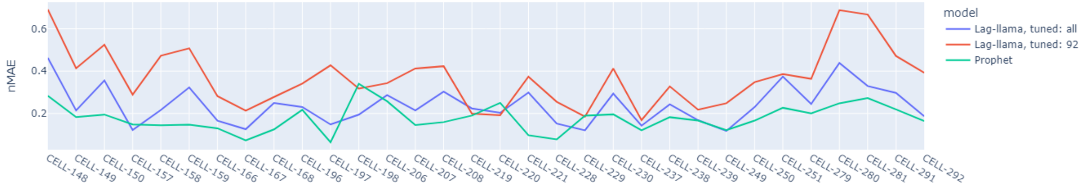


Figure 5.37: *The comparison of the Lag-Llama and Prophet models accuracy (nMAE) for UEs long-term forecasting per cell (BAND-1).*

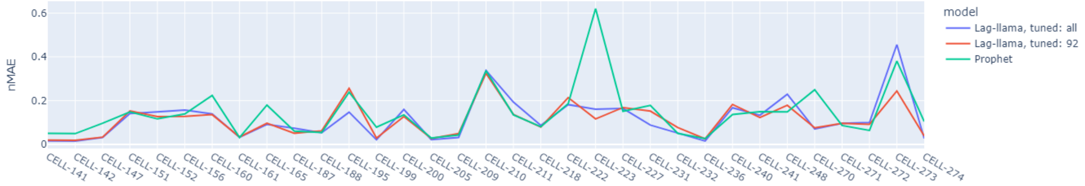


Figure 5.38: *The comparison of the Lag-Llama and Prophet models accuracy (nMAE) for UEs long-term forecasting per cell (BAND-2).*

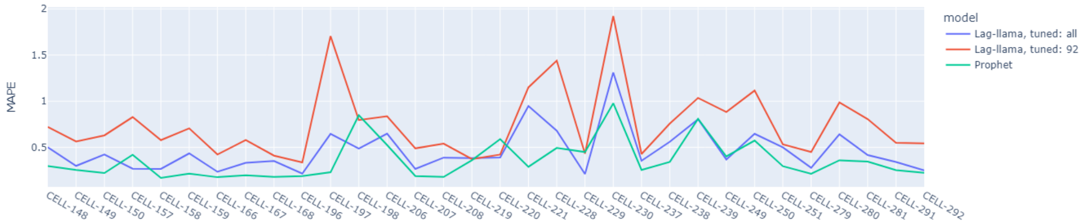


Figure 5.39: *The comparison of the Lag-Llama and Prophet models accuracy (MAPE) for UEs long-term forecasting per cell (BAND-1).*

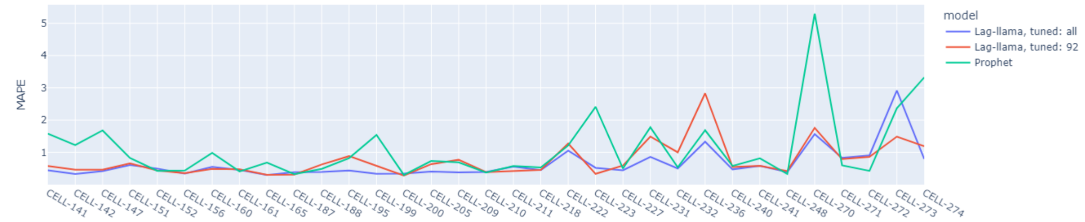


Figure 5.40: *The comparison of the Lag-Llama and Prophet models accuracy (MAPE) for UEs long-term forecasting per cell (BAND-2).*

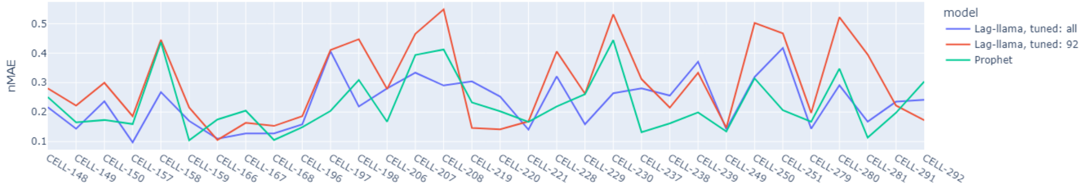


Figure 5.41: *The comparison of the Lag-Llama and Prophet models accuracy (nMAE) for CQI long-term forecasting per cell (BAND-1).*

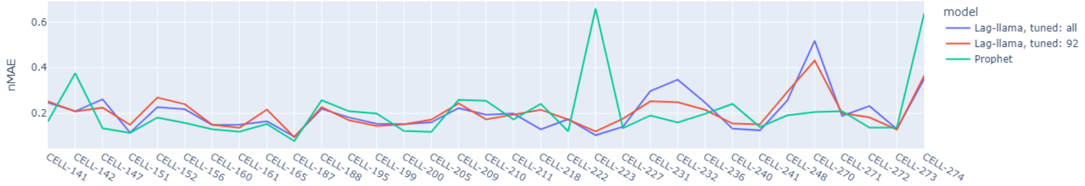


Figure 5.42: *The comparison of the Lag-Llama and Prophet models accuracy (nMAE) for CQI long-term forecasting per cell (BAND-2).*

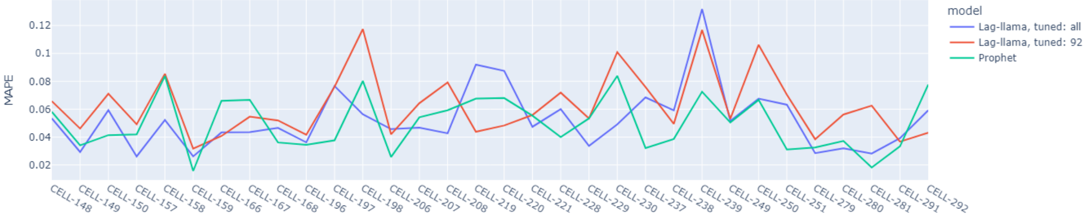


Figure 5.43: *The comparison of the Lag-Llama and Prophet models accuracy (MAPE) for CQI long-term forecasting per cell (BAND-1).*

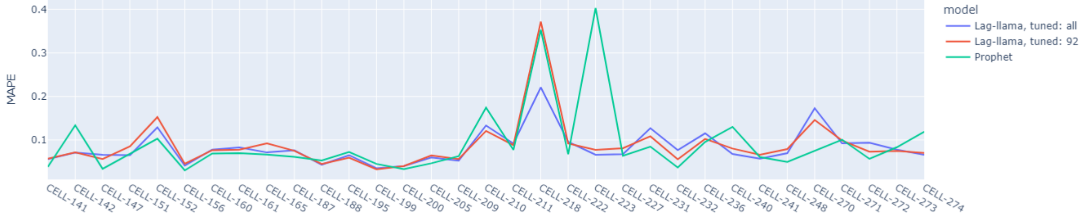


Figure 5.44: *The comparison of the Lag-Llama and Prophet models accuracy (MAPE) for CQI long-term forecasting per cell (BAND-2).*

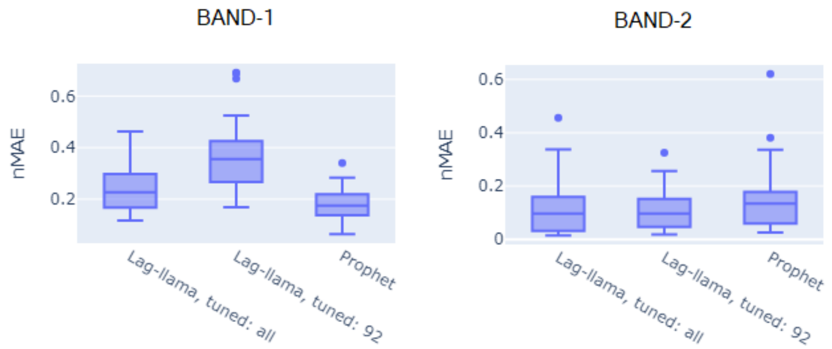


Figure 5.45: *The box plots of the Lag-Llama and Prophet models accuracy (nMAE) for UEs long-term forecasting.*

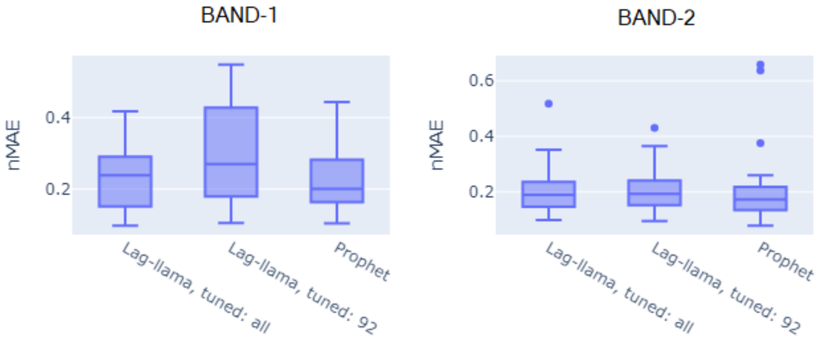


Figure 5.46: *The box plots of the Lag-Llama and Prophet models accuracy (nMAE) for CQI long-term forecasting.*

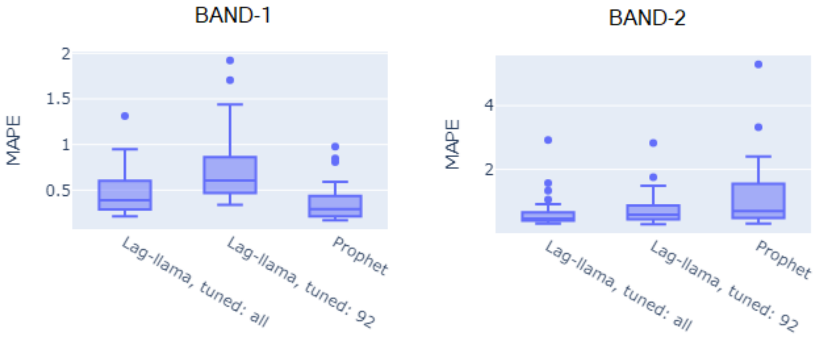


Figure 5.47: *The box plots of the Lag-Llama and Prophet models accuracy (MAPE) for UEs long-term forecasting.*

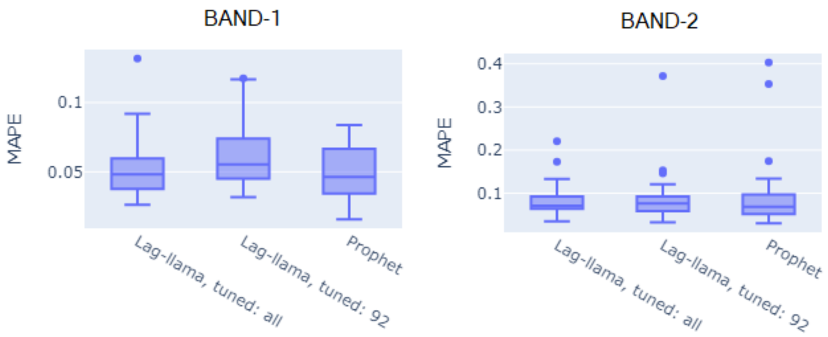


Figure 5.48: *The box plots of the Lag-Llama and Prophet models accuracy (MAPE) for CQI long-term forecasting.*

5.4 Summary

In the initial selection of models, SARIMA and Chronos have been ruled out from further research, and fine-grained analysis has been done for Lag-Llama and Prophet. Although zero-shot forecasts using Lag-Llama are notably less accurate than those with Prophet, the results for tuned versions of both Lag-Llama and Prophet are quite similar. In particular for CQI, both models deliver comparable outcomes and can be used interchangeably. However, for #UEs forecasting, Prophet provides more precise results for BAND-1, whereas Lag-Llama performs better for BAND-2. This indicates that signal characteristics play a crucial role (BAND-1 exhibits a stronger seasonal component). Both models perform similarly when trends are straightforward and seasonality is marked. However, for signals that exhibit more irregularities, a fine-tuned Lag-Llama outperforms.

Prophet stands out for its simplicity and ease of implementation, which is a significant advantage, but Lag-Llama offers great potential for fine-tuning, as it can leverage more data for learning (the training set used for long-term forecasting was limited and should be expanded in the future). Considering all these factors, it was decided to continue with Lag-Llama for further research involving multidimensional models.

Chapter 6

5G Short-Term Forecasting

This chapter assesses various data-driven models for short-term forecasting of slice-level throughput and delay. Using a multivariate approach, it integrates cell-specific radio and traffic conditions to offer accurate forecasts for each cell, achieving the highest level of detail in configuration. This approach will be applied during the dimensioning and planning stages of natural network evolution, thus eliminating the necessity for a predefined traffic model.

For short-term forecasting, a multistep method was devised in which forecasts for the next 24 hours are based on the preceding 24 hours of data. The evaluation dataset consists of the last 48 hours of data, divided equally into two segments. The first segment serves as input for the forecast, while the second segment is used to validate its accuracy.

6.1 Unit Models per Network Slice

Initially, research was focused on creating a method to model throughput and delay individually for each network slice to reduce the complexity of the problem, called in this dissertation *unit models*. The structure of the model for each network slice is illustrated in Fig. 3.1.

6.1.1 Results of Forecasting with VARMAX

The entire process that includes the preparation and modeling of data with VARMAX detailed in Sec. 4.2.1 is presented here with exemplary results per cell. VARMAX model requires complete and weakly stationary data. Each element of a multidimensional time series is broken down using a straightforward method based on the moving average with a 24-hour period. The decomposition process for an illustrative delay trajectory for slice A is shown in Fig. 6.1.

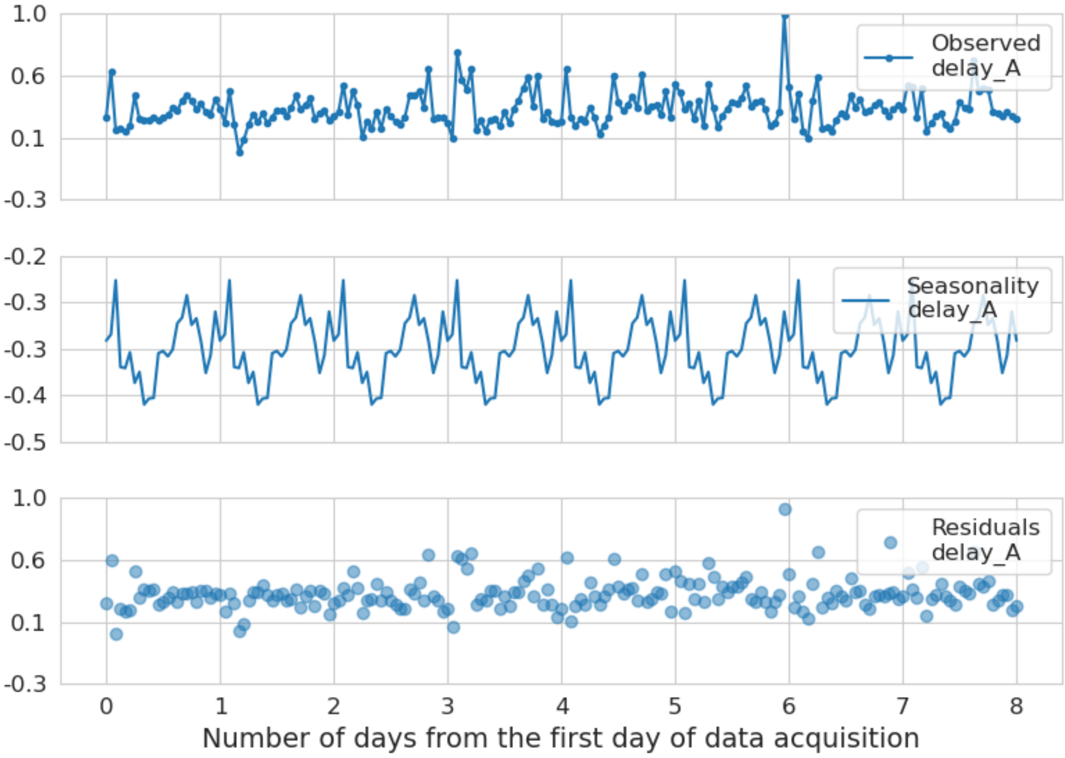


Figure 6.1: *Seasonal decomposition of delay for Slice A (BTS-10, CELL-303).*

Fig. 6.2 illustrates the Autocorrelation Function (ACF) for the same samples. Before decomposition, the data clearly exhibit seasonal patterns (Fig. 6.2A). After decomposition, these seasonal patterns in the ACF are no longer present (refer to Fig. 6.2B). This indicates that the decomposition process was effective.

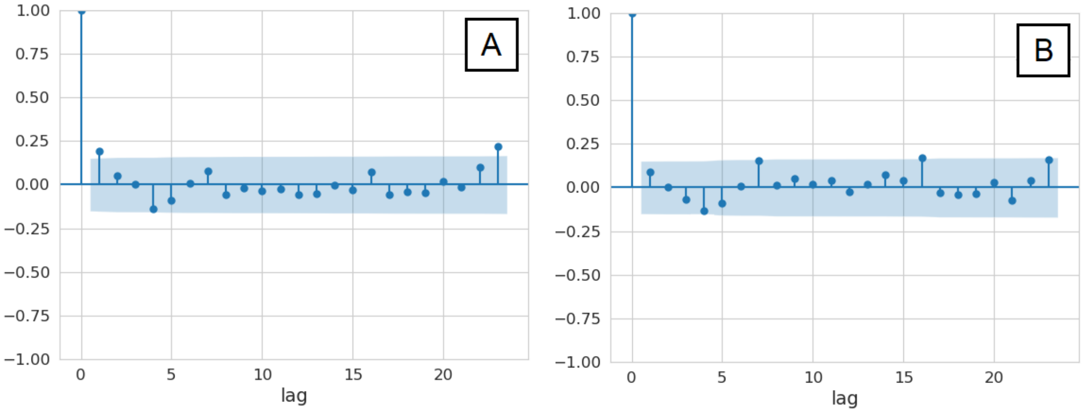


Figure 6.2: *Analysis of autocorrelation for delay (Slice A) for A: data before seasonal decomposition, B: data after seasonal decomposition (BTS-10, CELL-303).*

The next step following seasonal decomposition is to verify stationarity. If the time series is nonstationary, it should be differenced and this process should be repeated until stationarity is achieved. Stationarity is assessed using an Augmented Dickey-Fuller (ADF) test. Tab. 6.1 shows the results of the ADF test for an illustrative multidimensional time series corresponding to Slice A. As observed, the p-value for each variable is 0, indicating that the time series is stationary after seasonal decomposition. Hence, further differencing is unnecessary. However, it is important to note that differencing may be required for other cells.

Table 6.1: *Augmented Dickey-Fuller test (Slice A, BTS-10, CELL-303).*

Name	**Statistic**	**p-value**
TPut	-10.758	0
delay	-8.084	0
PRB utilization	-7.852	0
BLER	-6.838	0
DV	-10.223	0
CQI	-5.625	0
#UEs	-7.895	0

To select the range of VARMAX model orders, one-dimensional modeling experiments have been performed. ARMA model has been adjusted to each variable with orders from 0 to 12. The best orders for the univariate model are selected using the AIC, BIC and HQIC (Tab. 6.2). Because none of the orders of the univariate models exceeds 3 (for both moving average and autoregressive parts), the following orders have been selected for the VARMAX modeling: $(p, q) = (i, j)$, where $i, j \in \{0, 1, 2, 3\}$ and $(i, j) \neq (0, 0)$. The findings reveal that the lowest values of all the criteria are associated with $(p, q) = (1, 0)$. Therefore, this model was selected to forecast the delay and throughput of Slice A in this particular cell.

The prediction accuracy within the test set shows satisfactory results for both delay and throughput (Fig. 6.3 and Fig. 6.4). This is also the case for other network slices, indicated by the minimal normalized error values (Tab. 6.3). In this instance, the model order remains consistent across different network slices, though it may vary with other cells and slices.

Table 6.2: *Order selection of VARMAX(p, q) for Slice A, BTS-10, CELL-303.*

(*p*, *q*)	**AIC**	**BIC**	**HQIC**
(1, 0)	11667.00	11842.27	11738.13
(2, 0)	11671.36	11924.88	11774.25
(3, 0)	11675.53	12007.30	11810.17
(1, 1)	11712.78	11966.30	11815.66
(2, 1)	11716.06	12047.83	11850.70
(3, 1)	11726.56	12136.58	11892.95
(0, 1)	11731.12	11906.39	11802.25
(2, 2)	15797.14	16207.15	15963.53
(0, 2)	18737.30	18990.82	18840.18
(1, 2)	19383.97	19715.74	19518.61
(2, 3)	21171.69	21659.95	21369.84

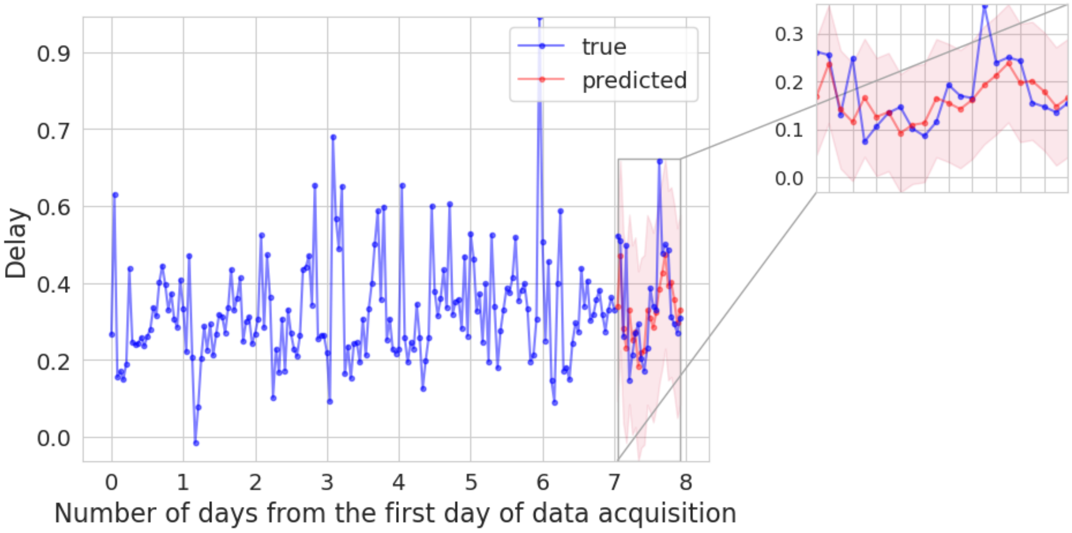


Figure 6.3: *The forecasted normalized delay for Slice A (BTS-10, CELL-303) utilizing the VARMAX model(1,0).*

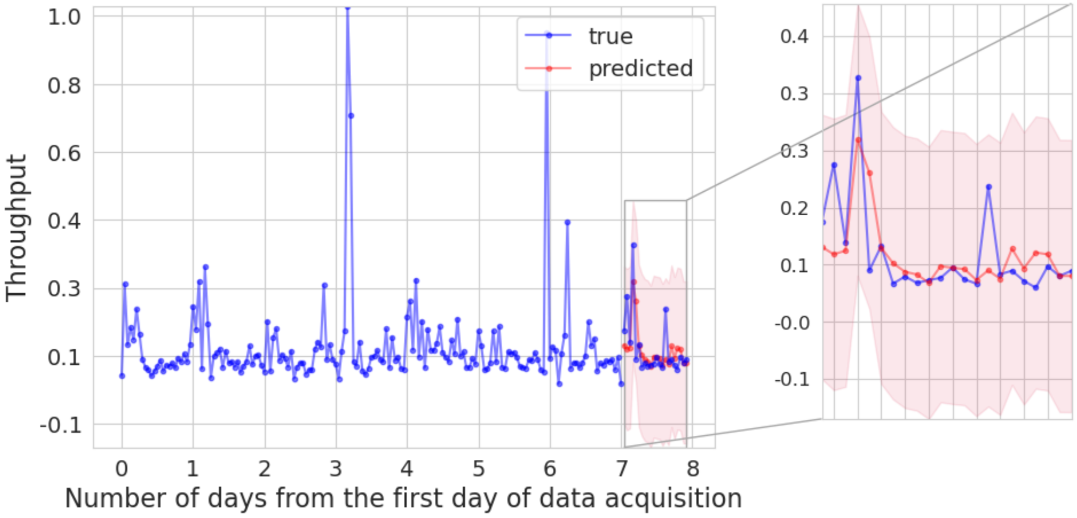


Figure 6.4: *The forecasted normalized throughput for Slice A (BTS-10, CELL-303) utilizing the VARMAX model(1,0).*

Table 6.3: *Evaluation metrics for example cell (BTS-10, CELL-303).*

Variable	QoS	(p,q)	nMAE	nRMSE	Comp. Time (s)	Range
Delay	A	(1, 0)	0.081	0.110	4.876	11076.877
	B	(1, 0)	0.096	0.141	5.119	20109.857
	C	(1, 0)	0.180	0.206	5.455	15097.226
	D	(1, 0)	0.129	0.162	6.090	13055.410
TPut	A	(1, 0)	0.044	0.070	4.876	55.596
	B	(1, 0)	0.100	0.186	5.119	35.686
	C	(1, 0)	0.118	0.181	5.455	129.242
	D	(1, 0)	0.069	0.091	6.090	127.936

The results presented until now are for data that are seasonally decomposed and differentiated (if necessary). As was mentioned before, reducing the number of variables can be beneficial, because it could make estimation of parameters faster. Therefore, both options for input data preparation for VARMAX modeling have been considered - with and without PCA (Tab. 6.4). Fig. 6.5 shows the comparison of time for both options. It is evident that the execution time for option VARMAX with PCA, which takes into account performing PCA, modeling and forecasting, is shorter for all slices.

Table 6.4: *VARMAX modeling options [77].*

Option	PCA	Endogenous	Exogenous
1	False	throughput, delay, data_volume, BLER, PRB utilization	#UEs, CQI
2	True	throughput, delay, PCA_1, PCA_2	#UEs, CQI

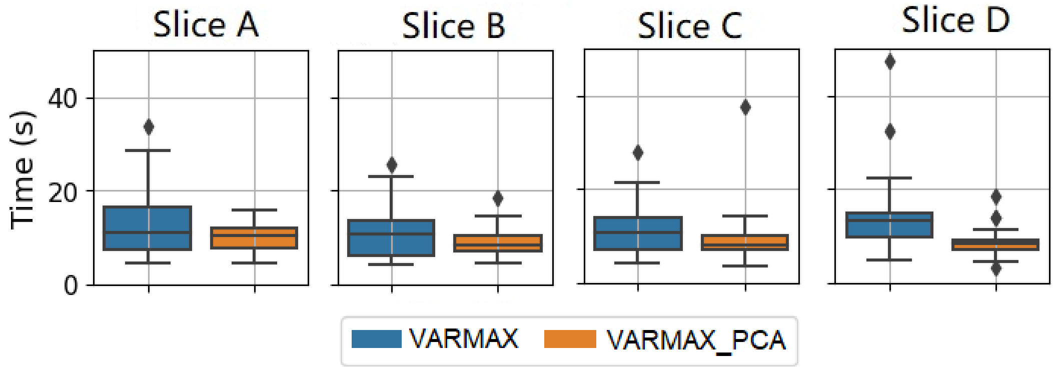


Figure 6.5: *Comparison of execution time between VARMAX and VARMAX PCA [77].*

In addition, the values of nMAE calculated for all cells are similar for both options (for both throughput and delay). This can be seen in the boxplots of nMAE presented in Sec. 6.1.3 that is discussing the comparative results in more detail. Therefore, employing the algorithm with PCA could be beneficial for practical purposes because of its rapid computational speed, yet its reduced interpretability must be considered.

6.1.2 Results of Forecasting with Neural Networks

For neural networks, the data is partitioned similarly to the VARMAX model. The selection of hyperparameters (i.e., configurations detailed in Sec. 4.2.2) is determined through supplementary testing. At first, an extensive range of hyperparameters is examined, although with smaller datasets. Of the original 24 different combinations (Tab.

6.5), the list was narrowed to 6 combinations, chosen as the best set of hyperparameters for further investigation: 1, 3, 5, 7, 9, 11.

Table 6.5: *LSTM unit configurations for all combinations of hyperparameter values [95].*

Model	units	Learning rate	Optimizer	Loss
1	[50]	0.01	adam	mse
2	[50]	0.01	sgd	mse
3	[50]	0.01	adam	mae
4	[50]	0.01	sgd	mae
5	[50]	0.001	adam	mse
6	[50]	0.001	sgd	mse
7	[50]	0.001	adam	mae
8	[50]	0.001	sgd	mae
9	[50, 50]	0.01	adam	mse
10	[50, 50]	0.01	sgd	mse
11	[50, 50]	0.01	adam	mae
12	[50, 50]	0.01	sgd	mae
13	[50, 50]	0.001	adam	mse
14	[50, 50]	0.001	sgd	mse
15	[50, 50]	0.001	adam	mae
16	[50, 50]	0.001	sgd	mae
17	[50, 50, 20]	0.01	adam	mse
18	[50, 50, 20]	0.01	sgd	mse
19	[50, 50, 20]	0.01	adam	mae
20	[50, 50, 20]	0.01	sgd	mae
21	[50, 50, 20]	0.001	adam	mse
22	[50, 50, 20]	0.001	sgd	mse
23	[50, 50, 20]	0.001	adam	mae
24	[50, 50, 20]	0.001	sgd	mae

After further research on the limited number of models for the same exemplary cell as discussed in Sec. 6.1.1, Model 3 demonstrates superior performance by achieving the lowest values for nMAE, normalized Root Mean Square Error (RMSE) (Tab. 6.6) and sum of the training and forecasting times. The forecasts for throughput and delay for Slice A are illustrated in Fig. 6.6 and Fig. 6.7.

Table 6.6: *Comparison of various neural network structures based on LSTM units. Evaluation metrics calculated for Slice A (BTS-10, CELL-303).*

Variable	Range	Model	nMAE	nRMSE	Comp. Time (s)
Delay	55.60	1	0.085	0.118	199.985
		3	0.076	0.115	280.170
		5	0.087	0.106	128.451
		7	0.055	0.078	115.143
		9	0.066	0.084	424.796
		11	0.069	0.112	205.252
TPut	11076.88	1	0.110	0.151	199.985
		3	0.106	0.143	280.170
		5	0.110	0.138	128.451
		7	0.103	0.129	115.143
		9	0.092	0.122	424.796
		11	0.096	0.137	205.252

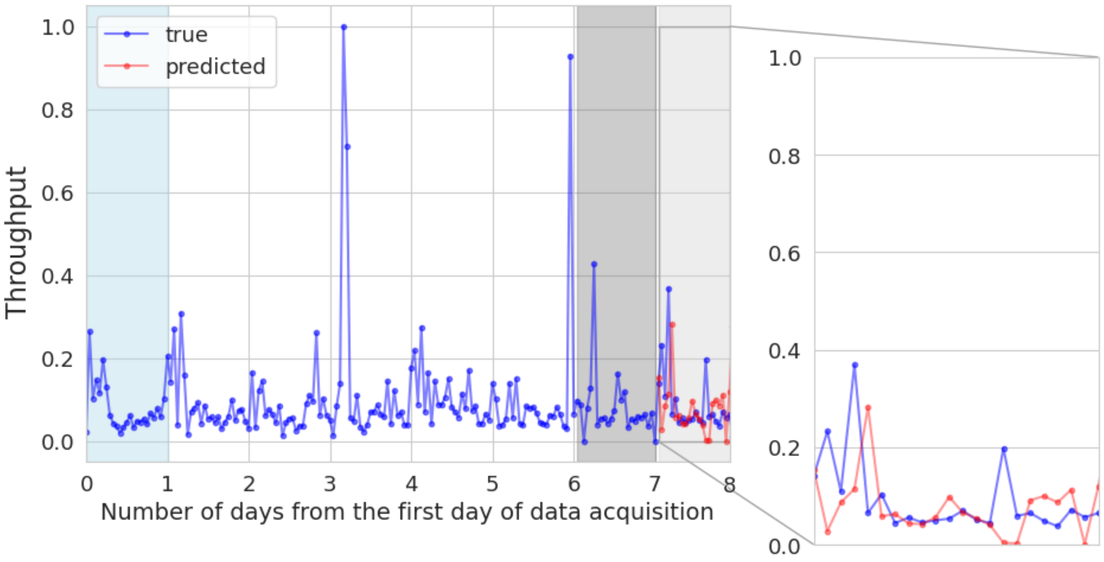


Figure 6.6: *Throughput forecast for Slice A (BTS-10, CELL-303) using Model 3.*

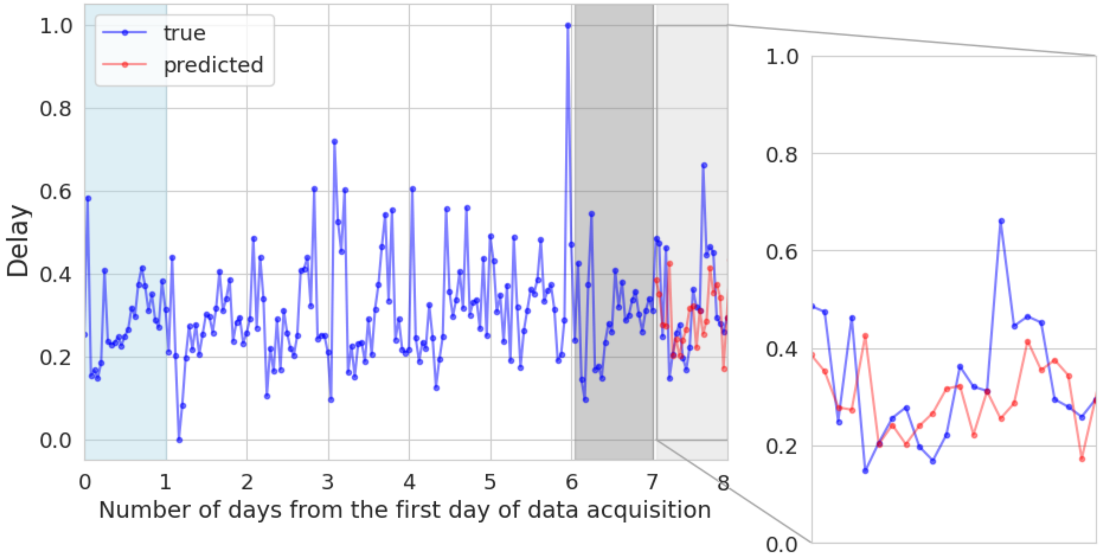
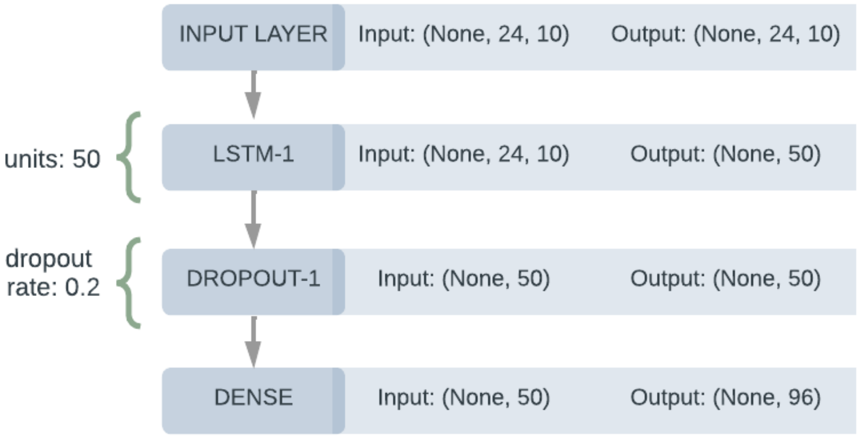


Figure 6.7: *Delay forecast for Slice A (BTS-10, CELL-303) using Model 3.*

The best model comprises a single hidden layer (Fig. 6.8), utilizing Adam as the optimizer [96, 97], Mean Absolute Error (MAE) as the loss function, and a learning rate set at 0.01 (Tab. 6.7). This configuration is selected as the most precise due to the fact that the MAE and Mean Squared Error (MSE) values are the smallest in most of the samples.



INPUT LAYER	Input: (None, 24, 10)	Output: (None, 24, 10)
LSTM-1	Input: (None, 24, 10)	Output: (None, 50)
DROPOUT-1	Input: (None, 50)	Output: (None, 50)
DENSE	Input: (None, 50)	Output: (None, 96)

Figure 6.8: *The architecture of the neural network incorporating LSTM units [77]. The values given in the brackets describe input shape. None means arbitrary number.*

Table 6.7: *Selected hyperparameters for LSTM block structures [77].*

Hyperparameter name	**The best structure**
LSTM-number of layers	1
LSTM-units	[50]
LSTM-dropout rate	0.2
LSTM-activation	tangent
LSTM-recurrent activation	sigmoid
Learning rate	0.01
Optimizer	Adam
Batch size	24
Loss	MAE

A similar analysis has been performed with the same steps has been conducted for Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM). All the considered hyperparameter combinations are listed in Tab. 6.8. An additional combination has been tested for Model 1 with dropout rate 0, because this model presented the best accuracy.

Comparison of this model with other analogs is presented in the next section.

Table 6.8: *CNN-BiLSTM unit configurations for all combinations of hyperparameter values. Values for CONV1 and CONV2 represent (filters, kernel size) for first and second convolutional layer. *Dropout rate for Model 2: 0, for others 0.1.*

Model	CONV1, CONV2	CONV-act.	Optimizer	Learning rate	Loss
1	[(128, 4), (64, 2)]	relu	adam	0.001	mae
2*	[(128, 4), (64, 2)]	relu	adam	0.001	mae
3	[(128, 4), (64, 2)]	relu	adam	0.001	mse
4	[(128, 4), (64, 2)]	relu	sgd	0.001	mae
5	[(128, 4), (64, 2)]	relu	sgd	0.001	mse
6	[(128, 4), (64, 2)]	relu	adam	0.0001	mae
7	[(128, 4), (64, 2)]	relu	adam	0.0001	mse
8	[(128, 4), (64, 2)]	relu	sgd	0.0001	mae
9	[(128, 4), (64, 2)]	relu	sgd	0.0001	mse
10	[(128, 4), (64, 2)]	tanh	adam	0.001	mae
11	[(128, 4), (64, 2)]	tanh	adam	0.001	mse
12	[(128, 4), (64, 2)]	tanh	sgd	0.001	mae
13	[(128, 4), (64, 2)]	tanh	sgd	0.001	mse
14	[(128, 4), (64, 2)]	tanh	adam	0.0001	mae
15	[(128, 4), (64, 2)]	tanh	adam	0.0001	mse
16	[(128, 4), (64, 2)]	tanh	sgd	0.0001	mae
17	[(128, 4), (64, 2)]	tanh	sgd	0.0001	mse
18	[(256, 4), (128, 4)]	relu	adam	0.001	mae
19	[(256, 4), (128, 4)]	relu	adam	0.001	mse
20	[(256, 4), (128, 4)]	relu	sgd	0.001	mae
21	[(256, 4), (128, 4)]	relu	sgd	0.001	mse
22	[(256, 4), (128, 4)]	relu	adam	0.0001	mae
23	[(256, 4), (128, 4)]	relu	adam	0.0001	mse
24	[(256, 4), (128, 4)]	relu	sgd	0.0001	mae
25	[(256, 4), (128, 4)]	relu	sgd	0.0001	mse
26	[(256, 4), (128, 4)]	tanh	adam	0.001	mae
27	[(256, 4), (128, 4)]	tanh	adam	0.001	mse
28	[(256, 4), (128, 4)]	tanh	sgd	0.001	mae
29	[(256, 4), (128, 4)]	tanh	sgd	0.001	mse
30	[(256, 4), (128, 4)]	tanh	adam	0.0001	mae
31	[(256, 4), (128, 4)]	tanh	adam	0.0001	mse
32	[(256, 4), (128, 4)]	tanh	sgd	0.0001	mae
33	[(256, 4), (128, 4)]	tanh	sgd	0.0001	mse

The best parameter setting for the CNN-BiLSTM architecture is shown in Fig. 6.9 and summarized in Tab. 6.9.

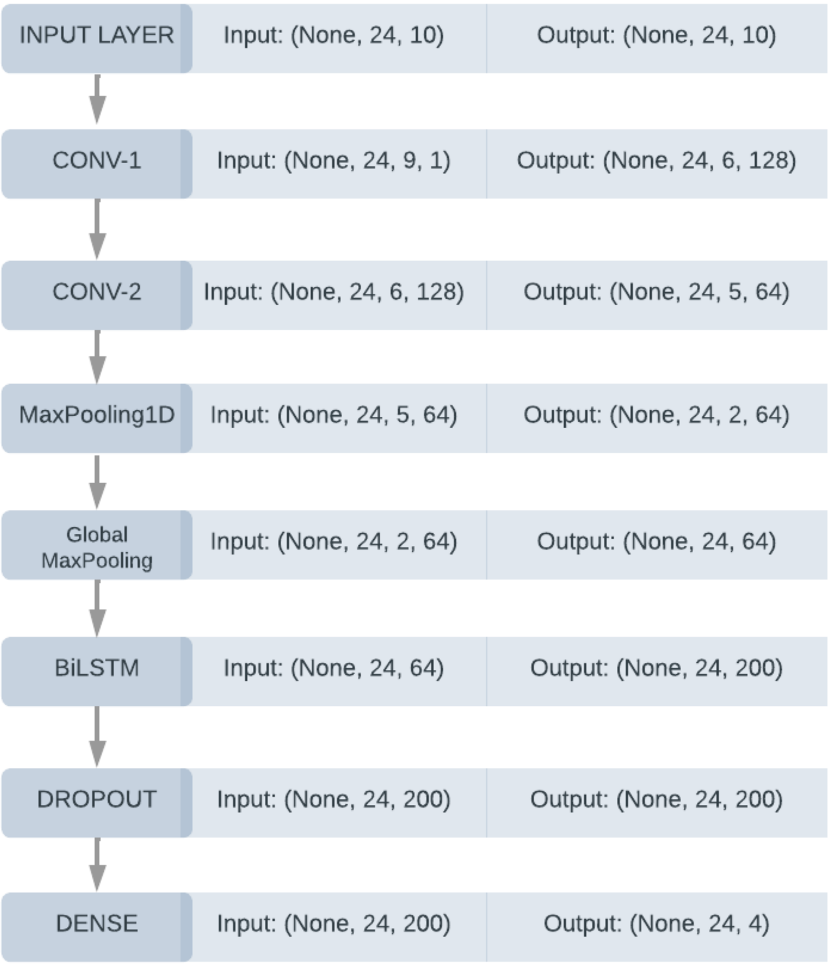


Figure 6.9: *The architecture of the neural network incorporating CNN-BiLSTM units [77]. The values given in the brackets describe input shape. None means arbitrary number.*

6.1.3 Comparative Study of Unit Models

After selecting the best setting of hyperparameters for each model, the comparative study of all unit models is done. For the example cell (BTS-10, CELL-303), the duration of processing and normalized errors for the best architectures of LSTM and CNN-BiLSTM were evaluated (Fig. 6.10 and 6.11). The processing duration is analyzed by summing up the learning and forecasting times, as learning must be performed individually for each cell, making it a significant component of the total processing time. It is evident that, for most of the network slices, the LSTM-based network requires the most training time, and VARMAX the least. The comparison of accuracy does not present such clear conclusions which may be due to the specifics of this singular analyzed cell.

Table 6.9: *Best hyperparameters for the CNN-BiLSTM architecture [77].*

Hyperparameter name	**The best structure**
CONV1 (filters, kernel size)	(128, 4)
CONV2 (filters, kernel size)	(64, 2)
CONV-activation	relu
Pooling size	2
BiLSTM-number of layers	1
BiLSTM-units	100
BiLSTM-activation	relu
BiLSTM-recurrent activation	relu
Dropout rate	None
Batch size	24
Learning rate	0.001
Optimizer	Adam
Loss	MSE

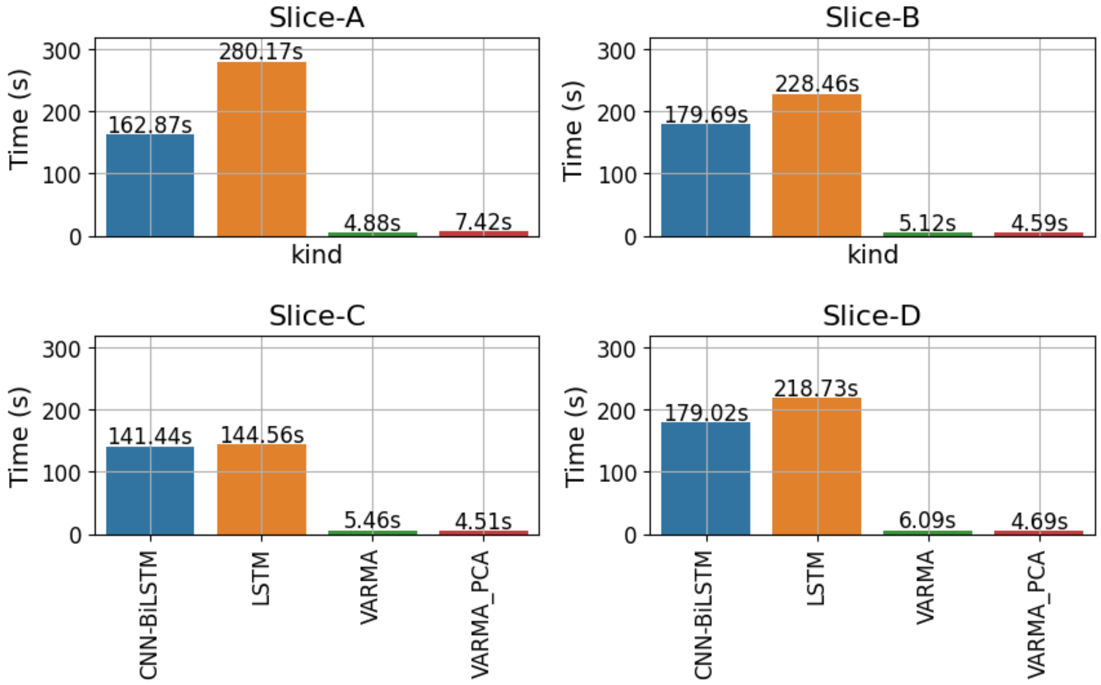


Figure 6.10: *Sum of learning and forecasting times compared for evaluated slice models (BTS-10, CELL-303).*

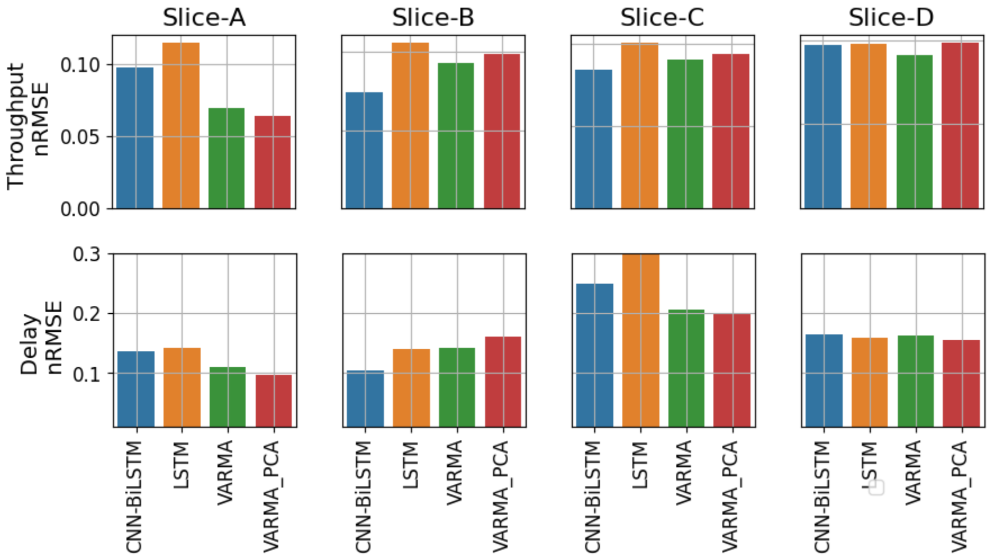


Figure 6.11: *Comparison of nRMSE per network slice (BTS-10, CELL-303).*

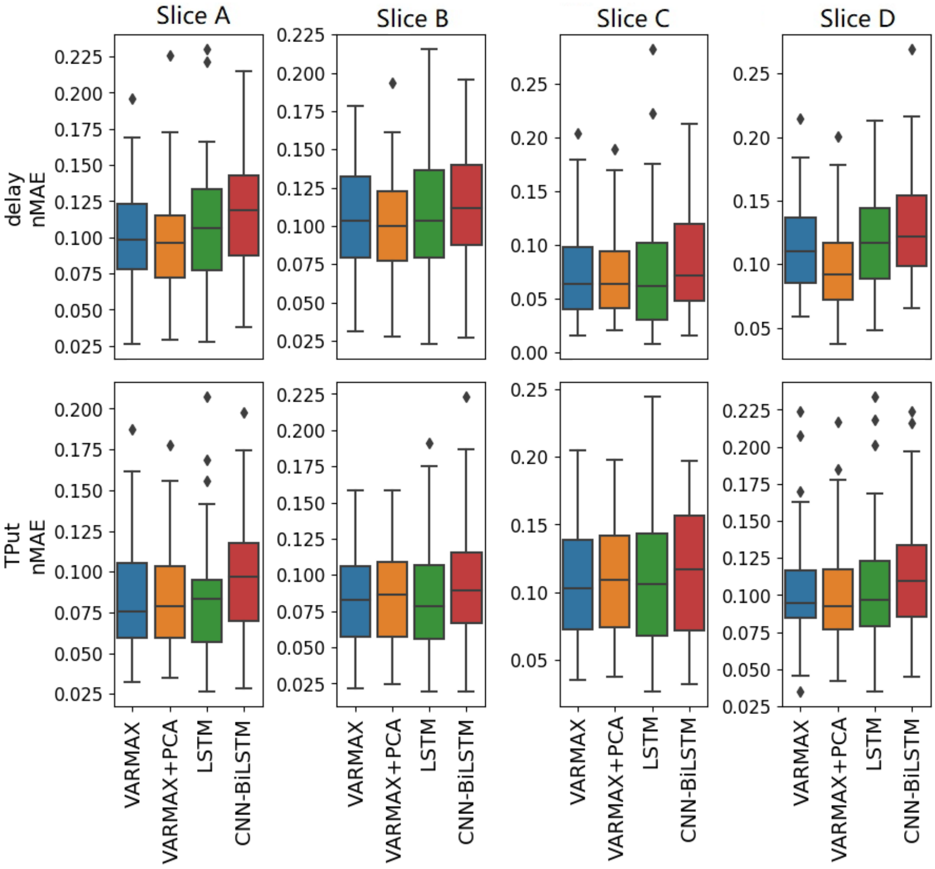


Figure 6.12: *The top panel shows the normalized Mean Absolute Error (MAE) for delay, while the bottom panel displays it for throughput across different network slices [77].*

The results obtained may vary according to the specific multivariate time series chosen, therefore, the evaluation metrics and duration of the modeling are examined for the remaining samples (and for each network slice) (Fig. 6.12 and Fig. 6.13).

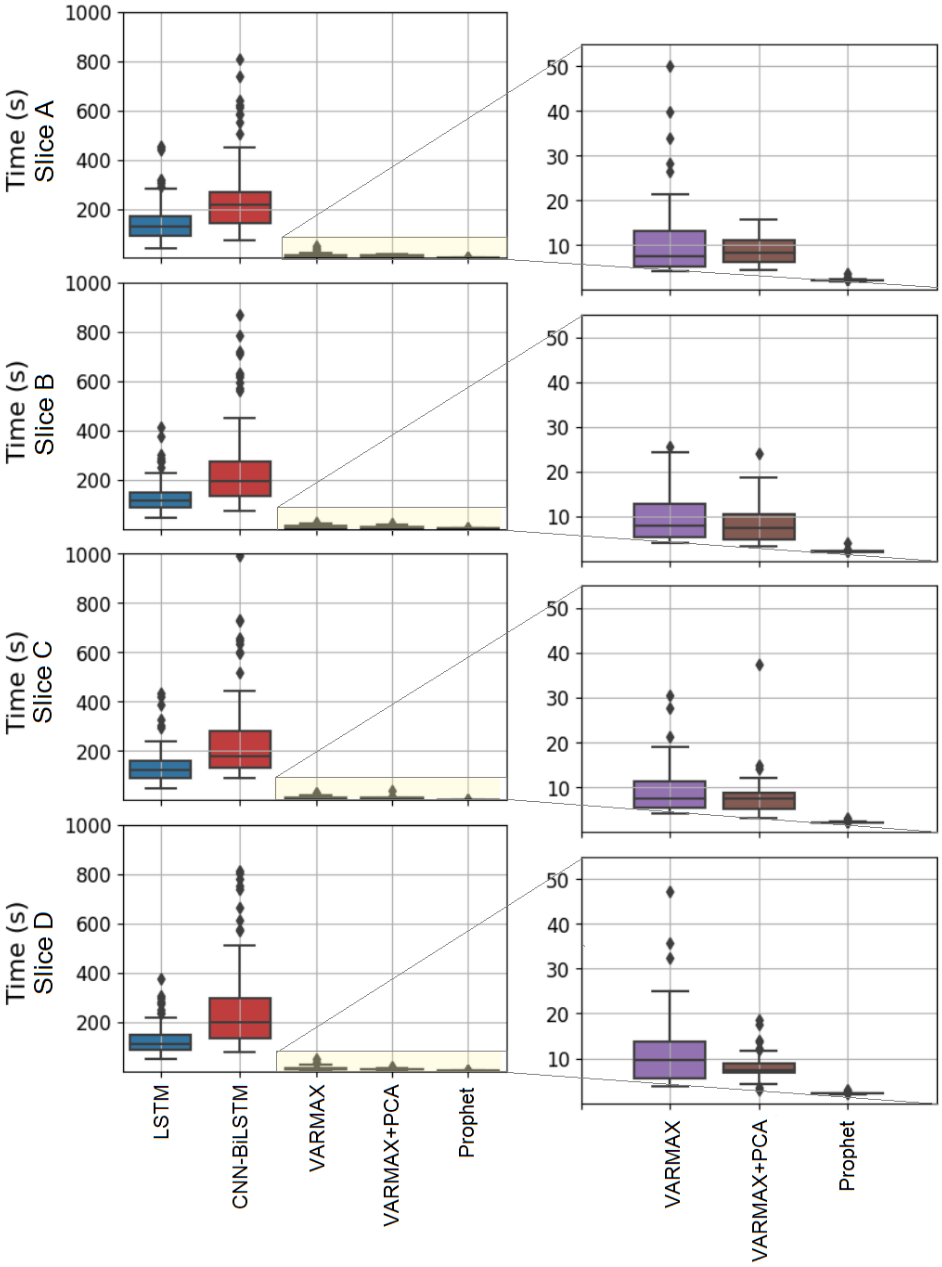


Figure 6.13: *Sum of learning and forecasting times compared for evaluated slice models (all cells) [77].*

The results presented in this section indicate that the VARMAX model (with or without PCA) may serve as a superior predictor with the best accuracy for most of the network slices and the shortest computational time. The forecast provided by the more intricate CNN-BiLSTM network is less precise. Furthermore, CNN-BiLSTM exhibits the longest duration for both modeling and forecasting.

6.2 General Model for All Network Slices

Considering the necessity to evaluate the effect of altering individual variables on traffic, a *general model* has been developed. This model encompasses data for all network slices within a particular cell. Although one method could involve combining unit models, a general model (incorporating variables for all network slices) is more straightforward to interpret. The schematic of the general model is shown in Fig. 6.14.

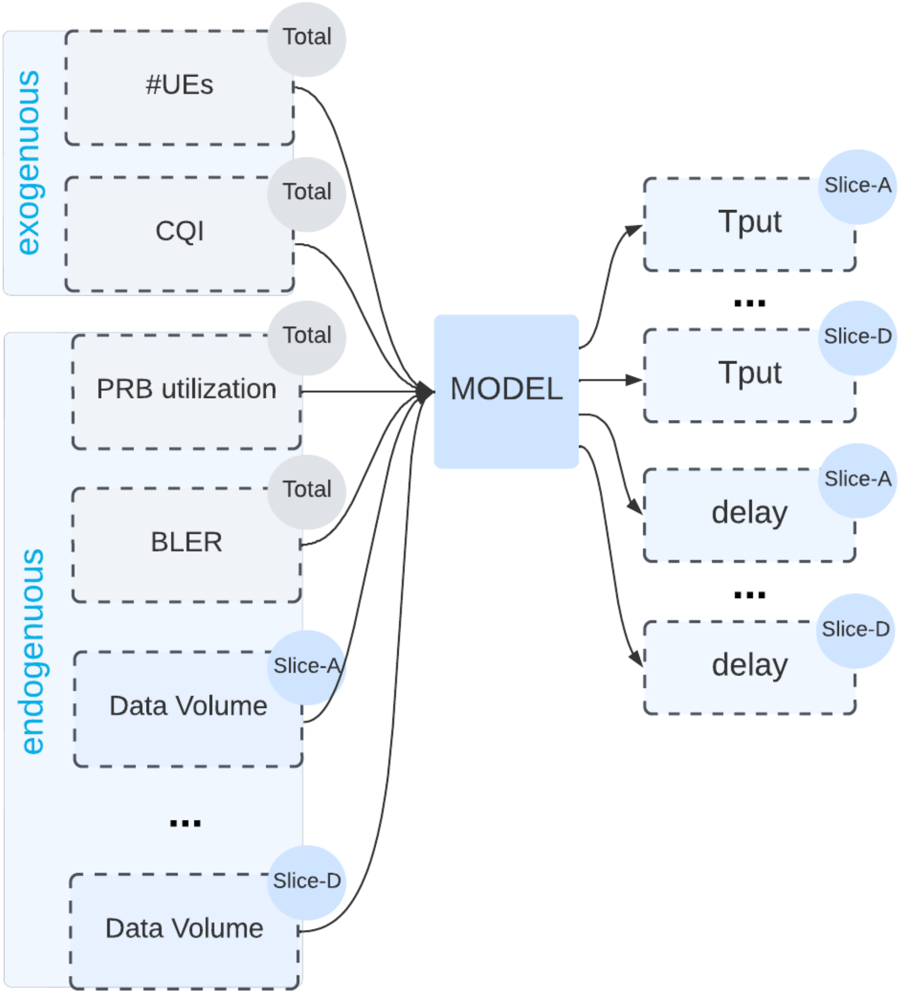


Figure 6.14: *Diagram of the general model with KPIs per each network slice.*

The time series decomposition and order selection process for the VARMAX model follows the same procedure as in Sec. 6.1. The method for selecting the best model remains unchanged; it aims to reduce the forecast error and, if desired, the computational time. Fig. 6.15 illustrates the count of "wins" for specific structures, which means the number of cells where certain models perform the best. The most precise LSTM-based neural network is Model 11, while the CNN-BiLSTM-based one is Model 2.

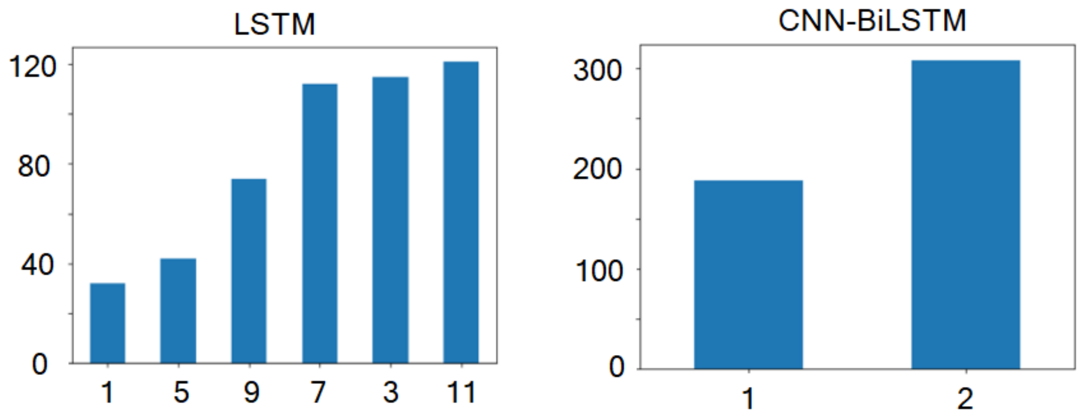


Figure 6.15: *The number of cells for which a specific structure yields the best performance.*

For a general model based on neural networks, the same structures as in Sec. 6.1 are evaluated. The lowest error for LSTM-based neural network is achieved by Model 11 and for the CNN-BiLSTM-based neural network by Model 2. The analysis of computational time for general model fitting and forecasting yields the same results as for unit models: VARMAX is the quickest, and the CNN-BiLSTM-based neural network is the slowest (Fig. 6.16).

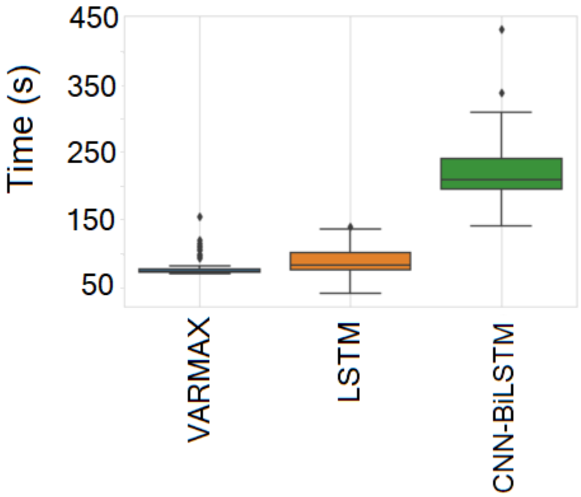


Figure 6.16: *Sum of learning and forecasting times compared for evaluated models (all slices and cells).*

Fig. 6.17 shows boxplots of normalized RMSE errors. The disparities between the general VARMAX model (G) and the unit model (U) are minimal for each network slice. However, this is not the case for LSTM-based models.

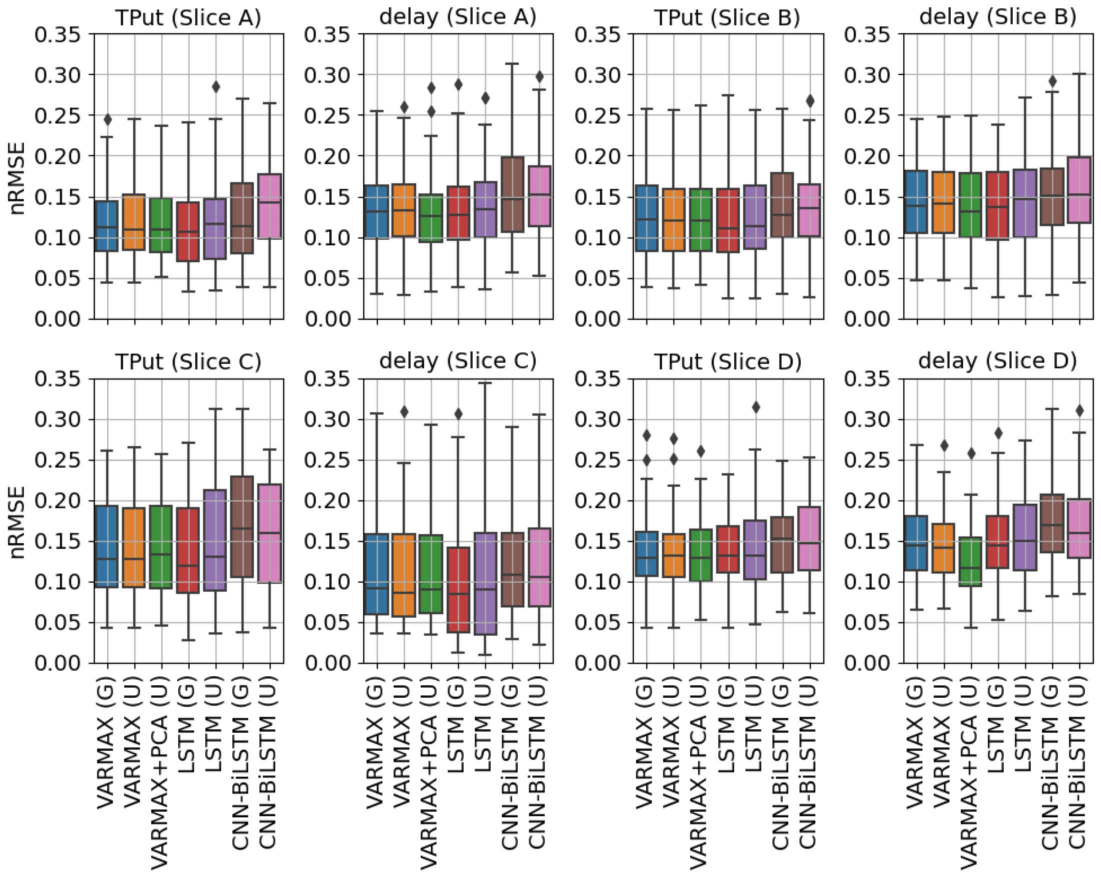


Figure 6.17: *Comparison of normalized RMSE for each slice. (G) represents the general model, and (U) represents the unit model.*

It is important to note that the best structures for unit and general models differ in their parameter sets (Model 3 is the best for unit, while Model 11 yields the best results for general models). Furthermore, the random selection of initial weights can influence network training. Typically, the lowest error is observed in various VARMAX models or LSTM-based networks. General models can effectively replace their individual counterparts. Moreover, in certain cases, VARMAX combined with PCA can produce the best forecasts. However, when dimensionality reduction is applied during modeling, the results become less interpretable. Depending on the desired functionality, one might consider whether to include PCA in the model.

It is not feasible to determine solely on visual inspection whether the differences in forecast errors between different models are statistically significant. To address this, the Friedman test and the Nemenyi *post hoc* test were conducted to determine when forecast errors differ significantly. In the field of statistics, the Nemenyi test serves as a *post-hoc* analysis to identify data groups that exhibit significant differences after a global test such as the Friedman test has rejected the null hypothesis, which asserts that the performance across data groups is similar. Essentially, if the p-value from the Friedman test falls below the predetermined significance threshold, it indicates that certain groups significantly differ, thereby justifying the use of the Nemenyi test [98]. The statistics and p-values for the Friedman test are shown in Tab. 6.10.

Table 6.10: *The Friedmann test to check if some groups differ significantly.*

Variable	**Slice**	**Statistic**	**p-value**
Throughput	A	78.35	7.81E-15
	B	73.17	9.15E-14
	C	56.93	1.88E-10
	D	41.38	2.43E-07
Delay	A	96.78	1.18E-18
	B	40.88	3.06E-07
	C	73.24	8.84E-14
	D	66.33	2.31E-12

The P-value is nearly 0 for every network slice, indicating that the null hypothesis can be dismissed. Consequently, for each network slice, there is a significant difference in at least one model's nMAE.

To verify the differences between pairs, the Nemenyi post hoc test is conducted. The findings for all slices are shown in Tab. 6.11 - 6.18. A p-value below 0.05 indicates a significant difference between the two algorithms at the 0.05 significance level. Otherwise, no statistically significant differences are found. The results for the other slices are similar, including for delay. General and unit models created by the same methods show no significant differences in any case. The most frequent differences are observed between CNN-BiLSTM (both U and G) and other methods. The forecast errors for the VARMAX model and the LSTM-based network are significantly different only for throughput in Slice A. For throughput and delay in Slice C the general LSTM model differs from VARMAX.

Table 6.11: *P-values from the Nemenyi post-hoc test. Variable: throughput, Slice A.*

	LSTM (G)	**LSTM (U)**	**CNN-BiLSTM (G)**	**CNN-BiLSTM (U)**	**VARMAX (G)**	**VARMAX (U)**	**VARMAX+PCA(U)**
LSTM (G)	1.000	0.018	0.001	0.001	0.008	0.001	0.009
LSTM (U)	0.018	1.000	0.248	0.001	0.900	0.900	0.900
CNN-BiLSTM (G)	0.001	0.248	1.000	0.064	0.387	0.900	0.361
CNN-BiLSTM (U)	0.001	0.001	0.064	1.000	0.001	0.001	0.001
VARMAX (G)	0.008	0.900	0.387	0.001	1.000	0.900	0.900
VARMAX (U)	0.001	0.900	0.900	0.001	0.900	1.000	0.900
VARMAX+PCA(U)	0.009	0.900	0.361	0.001	0.900	0.900	1.000

Table 6.12: *P-values from the Nemenyi post-hoc test. Variable: delay, Slice A.*

	LSTM (G)	**LSTM (U)**	**CNN-BiLSTM (G)**	**CNN-BiLSTM (U)**	**VARMAX (G)**	**VARMAX (U)**	**VARMAX+PCA(U)**
LSTM (G)	1.000	0.900	0.001	0.001	0.900	0.900	0.541
LSTM (U)	0.900	1.000	0.001	0.001	0.781	0.900	0.045
CNN-BiLSTM (G)	0.001	0.001	1.000	0.900	0.001	0.001	0.001
CNN-BiLSTM (U)	0.001	0.001	0.900	1.000	0.001	0.001	0.001
VARMAX (G)	0.900	0.781	0.001	0.001	1.000	0.900	0.661
VARMAX (U)	0.900	0.900	0.001	0.001	0.900	1.000	0.361
VARMAX+PCA(U)	0.541	0.045	0.001	0.001	0.661	0.361	1.000

Table 6.13: *P-values from the Nemenyi post-hoc test. Variable: throughput, Slice B.*

	LSTM (G)	**LSTM (U)**	**CNN-BiLSTM (G)**	**CNN-BiLSTM (U)**	**VARMAX (G)**	**VARMAX (U)**	**VARMAX+PCA(U)**
LSTM (G)	1.000	0.900	0.001	0.001	0.132	0.109	0.011
LSTM (U)	0.900	1.000	0.001	0.001	0.709	0.661	0.210
CNN-BiLSTM (G)	0.001	0.001	1.000	0.900	0.002	0.003	0.040
CNN-BiLSTM (U)	0.001	0.001	0.900	1.000	0.002	0.003	0.045
VARMAX (G)	0.132	0.709	0.002	0.002	1.000	0.900	0.900
VARMAX (U)	0.109	0.661	0.003	0.003	0.900	1.000	0.900
VARMAX+PCA(U)	0.011	0.210	0.040	0.045	0.900	0.900	1.000

Table 6.14: *P-values from the Nemenyi post-hoc test. Variable: delay, Slice B.*

	LSTM (G)	**LSTM (U)**	**CNN-BiLSTM (G)**	**CNN-BiLSTM (U)**	**VARMAX (G)**	**VARMAX (U)**	**VARMAX+PCA(U)**
LSTM (G)	1.000	0.806	0.001	0.016	0.878	0.854	0.900
LSTM (U)	0.806	1.000	0.007	0.440	0.900	0.900	0.613
CNN-BiLSTM (G)	0.001	0.007	1.000	0.661	0.004	0.005	0.001
CNN-BiLSTM (U)	0.016	0.440	0.661	1.000	0.361	0.387	0.045
VARMAX (G)	0.878	0.900	0.004	0.361	1.000	0.900	0.685
VARMAX (U)	0.854	0.900	0.005	0.387	0.900	1.000	0.661
VARMAX+PCA(U)	0.900	0.613	0.001	0.005	0.685	0.661	1.000

Table 6.15: *P-values from the Nemenyi post-hoc test. Variable: throughput, Slice C.*

	LSTM (G)	LSTM (U)	CNN-BiLSTM (G)	CNN-BiLSTM (U)	VARMAX (G)	VARMAX (U)	VARMAX+PCA(U)
LSTM (G)	1.000	0.098	0.001	0.001	0.071	0.021	0.001
LSTM (U)	0.098	1.000	0.001	0.088	0.900	0.900	0.709
CNN-BiLSTM (G)	0.001	0.001	1.000	0.806	0.001	0.007	0.132
CNN-BiLSTM (U)	0.001	0.088	0.806	1.000	0.121	0.290	0.878
VARMAX (G)	0.071	0.900	0.001	0.121	1.000	0.900	0.781
VARMAX (U)	0.021	0.900	0.007	0.290	0.900	1.000	0.900
VARMAX+PCA(U)	0.001	0.709	0.132	0.878	0.781	0.900	1.000

Table 6.16: *P-values from the Nemenyi post-hoc test. Variable: delay, Slice C.*

	LSTM (G)	LSTM (U)	CNN-BiLSTM (G)	CNN-BiLSTM (U)	VARMAX (G)	VARMAX (U)	VARMAX+PCA(U)
LSTM (G)	1.000	0.541	0.001	0.001	0.004	0.008	0.002
LSTM (U)	0.541	1.000	0.001	0.001	0.492	0.589	0.361
CNN-BiLSTM (G)	0.001	0.001	1.000	0.806	0.004	0.002	0.008
CNN-BiLSTM (U)	0.001	0.001	0.806	1.000	0.210	0.146	0.313
VARMAX (G)	0.004	0.492	0.004	0.210	1.000	0.900	0.900
VARMAX (U)	0.008	0.589	0.002	0.146	0.900	1.000	0.900
VARMAX+PCA(U)	0.002	0.361	0.008	0.313	0.900	0.900	1.000

Table 6.17: *P-values from the Nemenyi post-hoc test. Variable: throughput, Slice D.*

	LSTM (G)	LSTM (U)	CNN-BiLSTM (G)	CNN-BiLSTM (U)	VARMAX (G)	VARMAX (U)	VARMAX+PCA(U)
LSTM (G)	1.000	0.781	0.001	0.002	0.900	0.900	0.900
LSTM (U)	0.781	1.000	0.079	0.161	0.900	0.661	0.806
CNN-BiLSTM (G)	0.001	0.079	1.000	0.900	0.011	0.001	0.001
CNN-BiLSTM (U)	0.002	0.161	0.900	1.000	0.027	0.001	0.002
VARMAX (G)	0.900	0.900	0.011	0.027	1.000	0.900	0.900
VARMAX (U)	0.900	0.661	0.001	0.001	0.900	1.000	0.900
VARMAX+PCA(U)	0.900	0.806	0.001	0.002	0.900	0.900	1.000

Table 6.18: *P-values from the Nemenyi post-hoc test. Variable: delay, Slice D.*

	LSTM (G)	LSTM (U)	CNN-BiLSTM (G)	CNN-BiLSTM (U)	VARMAX (G)	VARMAX (U)	VARMAX+PCA(U)
LSTM (G)	1.000	0.661	0.004	0.007	0.900	0.900	0.071
LSTM (U)	0.661	1.000	0.361	0.440	0.781	0.210	0.001
CNN-BiLSTM (G)	0.004	0.361	1.000	0.900	0.009	0.001	0.001
CNN-BiLSTM (U)	0.007	0.440	0.900	1.000	0.014	0.001	0.001
VARMAX (G)	0.900	0.781	0.009	0.014	1.000	0.900	0.040
VARMAX (U)	0.900	0.210	0.001	0.001	0.900	1.000	0.387
VARMAX+PCA(U)	0.071	0.001	0.001	0.001	0.040	0.387	1.000

6.3 Summary

This chapter illustrates the effectiveness of the multidimensional VARMAX model in forecasting telecommunication data at various aggregation levels. It has been benchmarked against other forecasting techniques, such as LSTM, which exhibit high accuracy and efficient computational times. Improvements in previous short-term forecasting models have extended their applicability to forecasts to up to three months. Furthermore, training the model on data from all cells within a band, rather than just individual cells, has improved forecast accuracy. The findings indicate promising applications for the planning and scaling of 5G networks. Empirical evaluations using actual commercial network data have validated the practicality and reproducibility of the forecasting models and methodologies developed for long-term network slicing planning. Finally, this method has been integrated into the company's tools.

Chapter 7

5G Long-Term Forecasting

7.1 Long-Term Dataset Selection

For the purpose of long-term forecasting, the dataset utilized in Sec. 3.1 was reused. However, due to the requirement for extended periods of high-quality data (free of missing intervals, significant configuration changes, and with a non-zero traffic volume), it was further filtered. Consequently, the data, sourced from 33 active 5G BTS within a live network environment, were prepared and cleaned. The models were trained and tested using data that lasted 158 and 92 days, respectively. Subscribers in this network cluster were divided into three categories (one less than in the whole dataset due to low sporadic traffic in this slice):

- Slice A - mobile subscribers with high priority,
- Slice B - mobile subscribers with low priority,
- Slice C - fixed wireless access subscribers with lowest priority.

7.2 Forecasting Method

In Chapter 6, it was shown that the VARMAX model is an efficient method for short-term forecasting. This chapter extends the investigation to its long-term forecasting abilities. The VARMAX model, a vector ARMA model with additional exogenous inputs, is utilized.

The same procedure of hyperparameter selection has been used as in short-term forecasting, but with a longer dataset and at the BH level (Sec. 7.1, 3.6):

1. Data preprocessing with seasonal decomposition to ensure stationarity.
2. The determination of the model coefficients for the optimal order (p, q) with the MLE-based method.
3. Testing of various (p, q) pairs.
4. Selection of the one that minimizes the information criteria. The information criteria evaluated include the Akaike Information Criterion, the Bayesian Information Criterion, and the Hannan-Quinn Information Criterion.

7.3 Model per Cell

As discussed in Sec. 3.6, emphasis for long-term forecasting is on BH. Fig. 7.1 and Fig. 7.2 illustrate an example of delay and throughput forecasts for Slice C. It is evident that the model forecasts remain accurate, despite the complete transformation in signal characteristics (e.g., by smoothing seasonal trends during aggregation to BH).

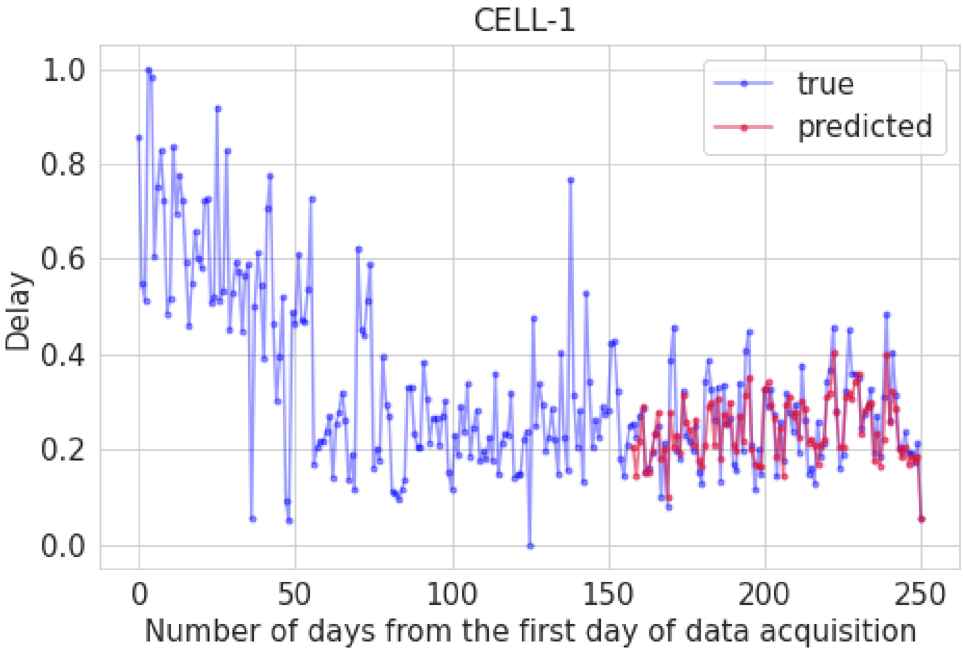


Figure 7.1: *The long-term forecast of normalized delay for Slice C utilizing VARMAX(1,0) for a representative cell(BTS-20, CELL-2).*

Tab. 7.1 displays the average normalized MAE for forecasts across the 185 analyzed cells, including those under high load conditions (Sec. 3.6). The high accuracy of the results for each slice indicates that this method is effective in forecasting delay and throughput, which are critical for slice QoS, over an extended period of around 3 months.

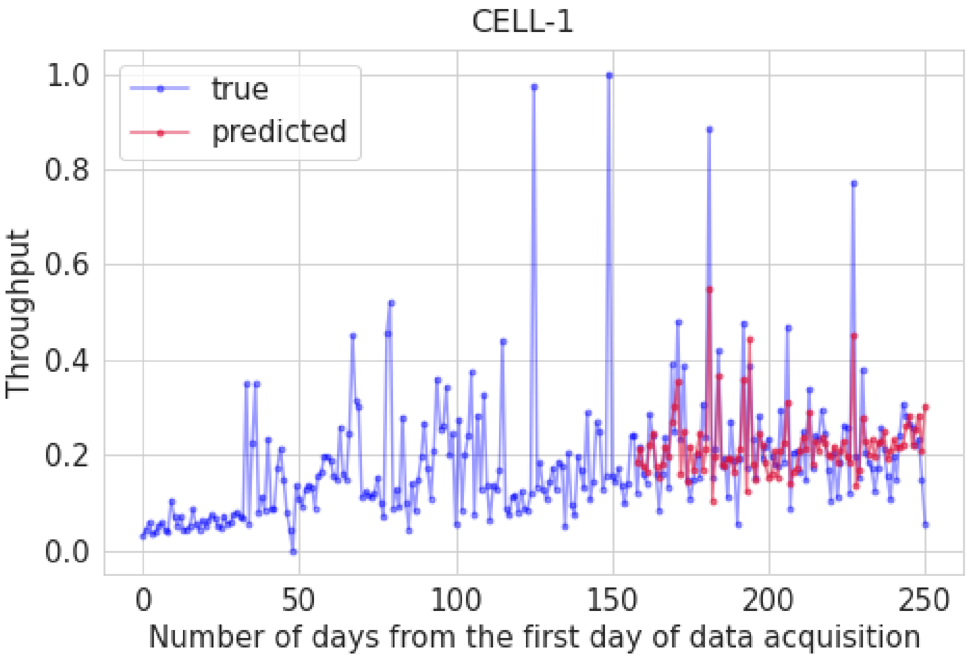


Figure 7.2: *The long-term forecast for normalized throughput for Slice C utilizing VARMAX(1,0) for a representative cell.*

Table 7.1: *The mean of nMAE (± standard deviation) for the long-term forecasts made for all and high-loaded cells.*

Variable	Cells	Slice A	Slice B	Slice C
Throughput	All	0.109 ± 0.045	0.112 ± 0.041	0.097 ± 0.036
	HL	0.127 ± 0.053	0.116 ± 0.041	0.073 ± 0.027
Delay	All	0.100 ± 0.043	0.094 ± 0.043	0.010 ± 0.046
	HL	0.098 ± 0.054	0.087 ± 0.040	0.087 ± 0.047

7.4 Model per Frequency Band

Training a model with historical data for each cell individually will result in a model that can accurately forecast its future values. However, if the data history is short or the cell traffic is not saturated, the data may be inadequate, leading to higher forecast errors. Consequently, a method for generalizing the model by training it on a larger set of cells has been explored. Based on expert domain knowledge, the frequency band and bandwidth have been selected as criteria to group cells for a unified model, as these criteria reflect the general conditions of the radio interface in terms of propagation characteristics and spectrum availability. The configuration combinations present in the dataset are shown in Tab. 3.2.

Building on the model developed for a single cell (Sec. 6.2), it was tested whether training the model on collected data from multiple cells within the same frequency band would yield a comparable forecast accuracy. The modeling procedure for the individual band is carried out as follows:

1. Take the cells that correspond to the particular band.

2. For each exogenous and endogenous variable, calculate the average value over time.

3. Fit the model to the trajectory obtained.

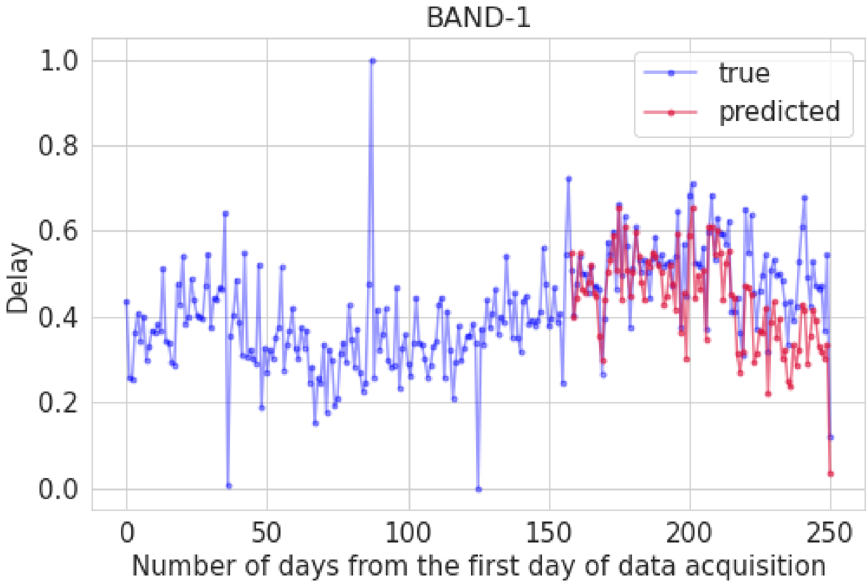


Figure 7.3: *The long-term forecast for normalized delay for Slice C using VARMAX(1,0) for BAND-1.*

As noted previously, cells with high PRB utilization are of particular interest. Forecasts for sample bands (where only heavily loaded cells were included in the modeling) are shown in Fig. 7.3 - 7.8. The forecasting model proves to be effective in both scenarios, despite the different data characteristics. Furthermore, it should be noted that band 1 (Fig. 7.3, Fig. 7.4) corresponds to the cell depicted in Fig. 7.1.

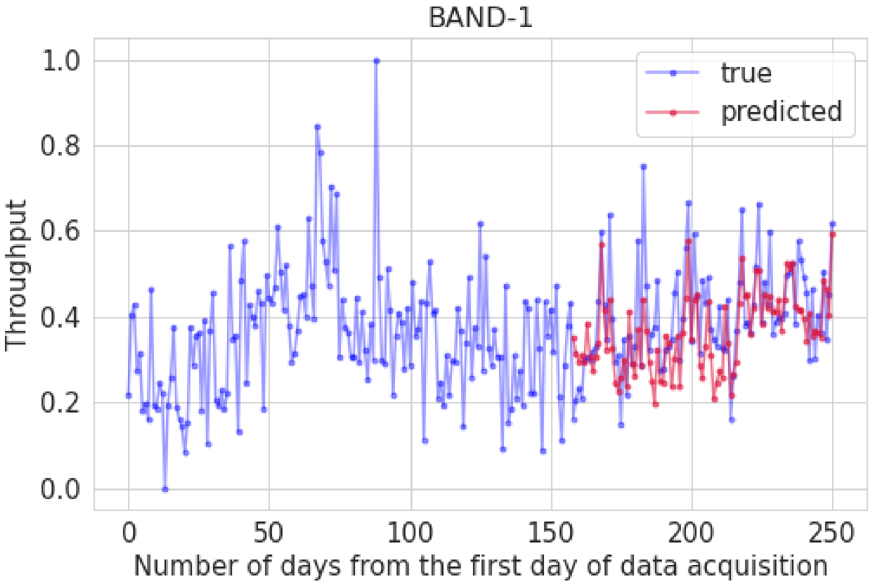


Figure 7.4: *The long-term forecast for normalized throughput for Slice C using VARMAX(1,0) for BAND-1.*

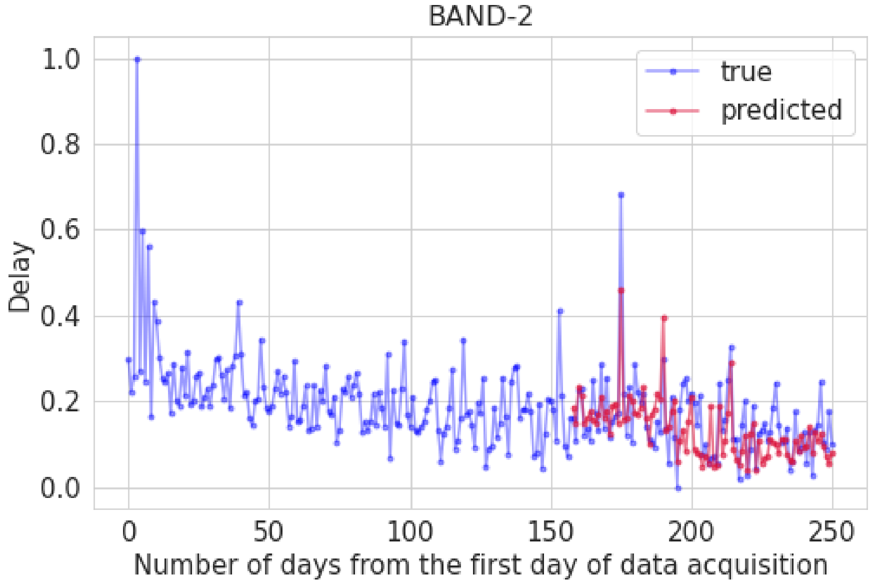


Figure 7.5: *The long-term forecast for normalized delay for Slice C using VARMAX(1,0) for BAND-2.*

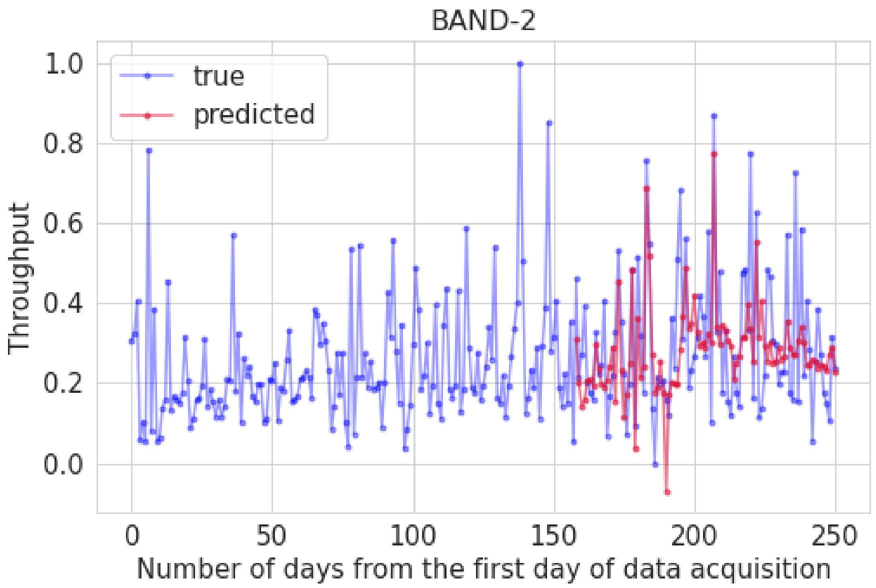


Figure 7.6: *The long-term forecast for normalized throughput for Slice C using VARMAX(1,0) for BAND-2.*

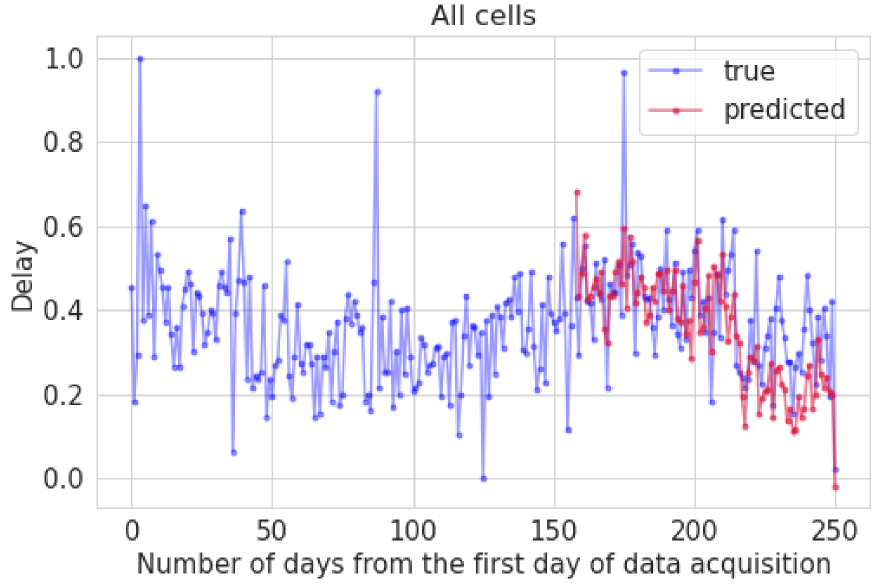


Figure 7.7: *The long-term forecast for normalized delay for Slice C using VARMAX(1,0) for all high-loaded cells in the network.*

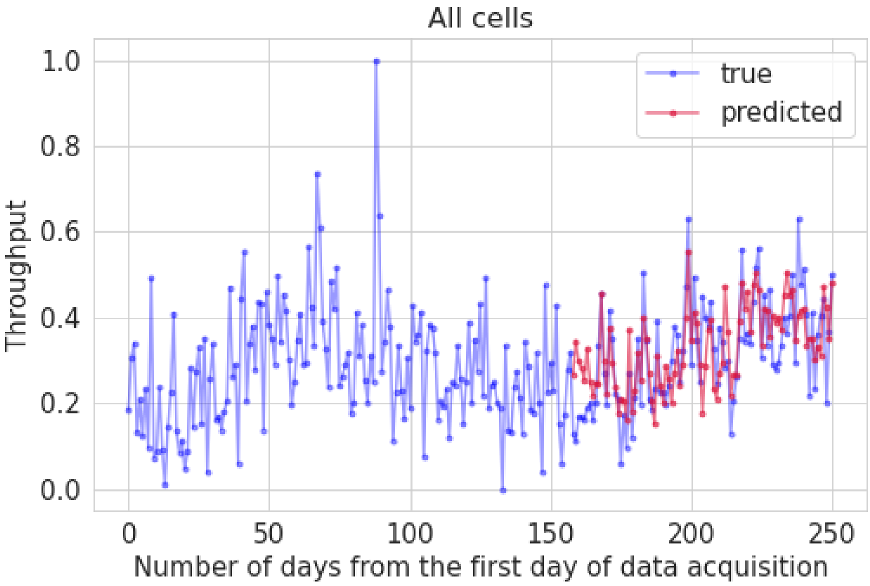


Figure 7.8: *The long-term forecast for normalized throughput for Slice C using VARMAX(1,0) for all high-loaded cells in the network.*

Table 7.2: *The nMAE corresponding to the long-term forecasts made with a model trained for each band and all cells.*

Variable	Cells	Slice A	Slice B	Slice C
Throughput	Band-1	0.209	0.141	0.073
	Band-2	0.122	0.093	0.114
	All	0.116	0.104	0.071
Delay	Band-1	0.097	0.085	0.075
	Band-2	0.088	0.053	0.050
	All	0.121	0.073	0.088

Tab. 7.2 presents the nMAE values for each slice and band. The error values vary slightly between bands, but the nMAE remain small and are generally lower than those of the per cell model and the model for all cells. Furthermore, it is important to note that the delay errors are smaller because the KPI that determines throughput is noisier, affecting the quality of model fitting.

7.5 Industrial Applications

The long-term throughput and delay forecasting model presented in this chapter has been integrated into a dashboard used for capacity analysis and dimensioning tasks at

Nokia. The method has been crafted using Microsoft Power BI [99]. The models are trained using data from all cells sharing the same band configuration as outlined in Sec. 7.4. The main objective of this dashboard is to evaluate the volume of traffic within the network cluster and to forecast when each cell will reach its capacity.

At the moment of writing, the dashboard is in the testing and fine tuning phase, consequently, there are no case studies that have been developed from it yet.

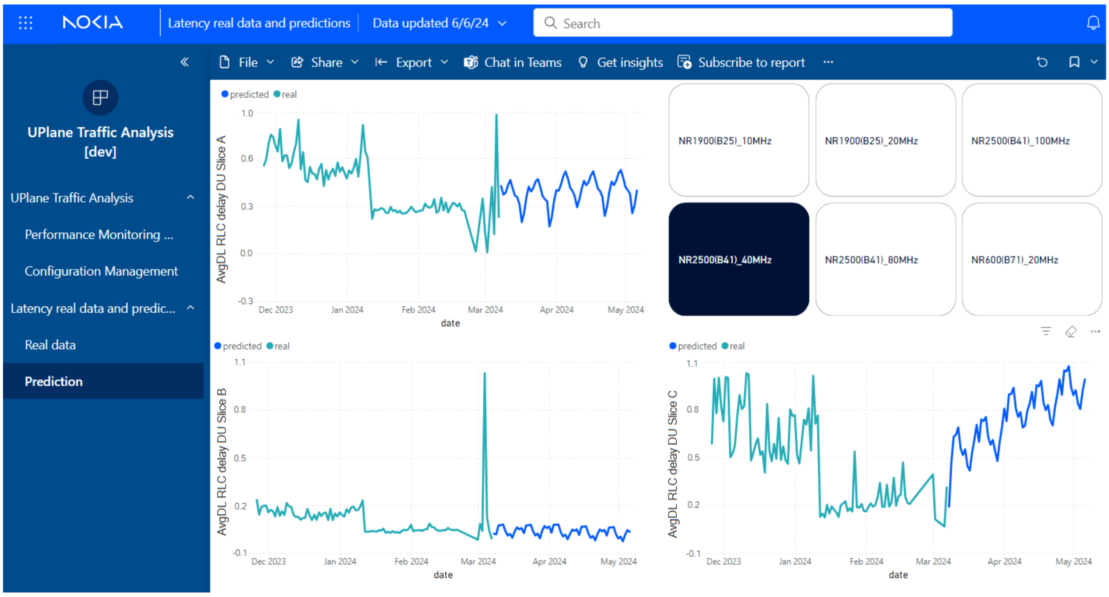


Figure 7.9: *Dashboard implementing long-term forecasts for real network data.*

7.6 Summary

The results detailed in this chapter indicate that the model chosen for short-term throughput and delay forecasting (Chapter 6) is equally effective for long-term predictions with commendable accuracy. After slight adjustments, specifically the transition from hourly time series to BH, the metrics demonstrate that the VARMAX model can accurately forecast 2-3 months into the future when trained on a 5-month dataset. Additionally, incorporating configurational knowledge, such as the frequency band, into the model's training process enhances its accuracy.

Chapter 8

Traffic Multiplexing Gain Estimation

The responsibility of Mobile Network Operators (MNOs) is to effectively plan, deploy, and maintain networks comprising numerous radio BTS. With the emergence of 5G and cloud network architecture, the complexity and number of interfaces that need to be planned are on the rise. Particularly in the domain of transport networks that interconnect all network components, MNOs must carefully address the requirements of fronthaul and midhaul links. In the era of 2G, 3G and 4G, the planning of fronthaul and midhaul links was not a major concern, as they were typically established as direct cable links connecting two network elements situated in close proximity, such as on the same site as a roof or an antenna mast (Sec. 1.3.5). Previous technologies mainly emphasized backhaul considerations. Transport planning plays a crucial role in the overall RAN planning since optimizing the cost of the transport network can result in reducing the overall expenses of deploying the mobile network. However, it is essential for transport links to offer adequate capacity and ensure QoS to support the required radio performance, which is typically included in the service commitments and marketing strategies of MNOs. Hence, there is a demand for cost reduction in the transport network, but not at the expense of compromising radio interface performance.

An aspect of the RAN planning process involves capacity dimensioning, which aims to estimate the resources needed to provide services to specific traffic while maintaining QoS. In the context of transport network dimensioning, this primarily involves estimating link capacities. Cloud-RAN technology divides the BTS into RU, DU, and CU, which can be physically separated to facilitate resource sharing. This requires MNOs to address both fronthaul and backhaul link capacity dimensioning. In the initial phase of 5G deployment, fronthaul link capacity is determined by the functional split between RU and DU in a consistent manner. In the subsequent phase, the fronthaul capacity will be influenced by the volume of traffic transmitted. Conversely, midhaul links handle packet-based traffic, and their capacity is only dictated by the traffic profile requirements. Moreover, the midhaul network can aggregate traffic from multiple cells in the network (Fig. 1.3, thus expanding the need for midhaul dimensioning on both the CU and DU ends. On the CU side, the midhaul link consolidates traffic flows from various DUs and offers potential cost savings through statistical MG, which is achievable due to the intermittent nature of packet traffic.

8.1 Statistical Multiplexing Gain

MG represents a measure of gain that can be obtained by sharing a transportation connection [100]. In the context of mobile transport networks, MG illustrates the variance in capacity needed for all cells combined at a specific location and the total traffic of these cells. In other words, it indicates the extent to which capacity at the aggregation point $Cap^{Agg.}$ can be decreased compared to the capacities required at each individual cell link $Cap^1, \ldots, Cap^N$ (where N denotes the number of cells). In practice, the sum of individual link capacities (estimated separately) is greater than the capacity of the aggregation link [67],[100] – refer to Equation 8.1. The corner case is when they are equal (which happens when there is no traffic or there is so-called full buffer traffic and all links are utilized in 100%) and $MG = 0$.

$$MG_P\,[\%] = \left(1 - \frac{Cap_P^{Agg.}}{\sum Cap_P^N}\right) \times 100\% \tag{8.1}$$

The primary factors influencing the gain in statistical multiplexing include the traffic pattern (such as the variability in traffic intensity) and the maximum data rates supported by the radio interface (Fig. 8.1). The variability in traffic intensity is determined by the traffic profile, which indicates the frequency and amount of data transfers needed by the UEs and the radio capacity available at that time. Hence, it is essential to examine various traffic profiles in conjunction with potential 5G radio setup elements to comprehend the MG in the midhaul.

Dimensioning is typically carried out on the basis of the peak resource demands, which occur during the BH (as was explained in Sec. 3.6). In addition to this, various factors influence the capacity at the aggregation point, such as the distribution of peak hours throughout the day and across different cells, as well as the movement of users within the network (which is not addressed in this research).

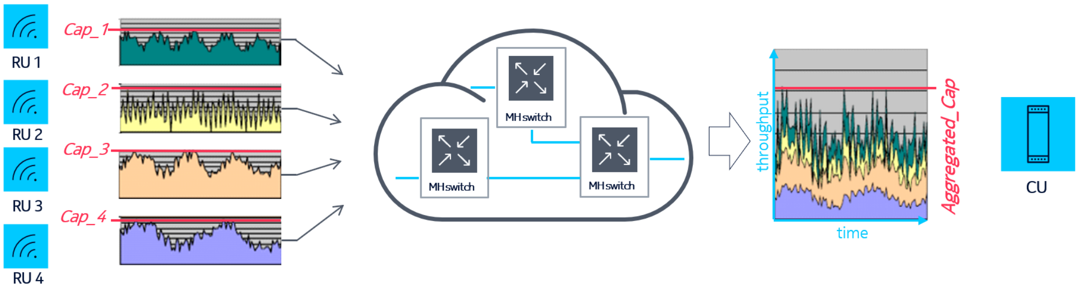


Figure 8.1: *Multiplexing gain concept.*

In this chapter, the instantaneous rates achievable under real system conditions at the radio interface are examined, as this provides an accurate benchmark of system performance and capabilities. The Cumulative Distribution Function (CDF) offers a range of possible throughputs for each link. The capacity of a link is determined by the CDF percentile (P) of achievable throughputs on that link. When $P = 1$, the transport capacity will be equal to the maximum rate or the sum of the maximum rates for fronthaul and midhaul links, respectively. A similar method was used in [21], where high delay percentiles were used as the main design metric instead of the more commonly used average delays in the existing literature. To minimize network expenses, CSPs can opt to restrict investments in the transport network in favor of the radio component. However, this cost reduction may affect radio performance, and the percentile indicates the extent of this impact. Consequently, each Cap_n and MG are determined as functions of P.

8.2 Simulation Setup

In order to analyze the QoS requirements (instantaneous data rates) of the wireless interface and traffic profile trends, a proprietary Nokia system level simulator has been used. That is, a fully dynamic system-level simulator implemented to simulate user plane performance, including throughputs and delays, using provided radio resource management features and algorithm variants. This tool allows for the assessment of different paths for the evolution of radio networks through performance and capacity simulations, considering specific scenarios with varying network setups, traffic loads, and signal propagation conditions. The network setup encompasses a detailed configuration of the physical layer and Radio Resource Management functionalities, including the generation of traffic patterns for multiple scenarios with user movement. Ultimately, it calculates the quality of the radio interface for users in the granularity of the Transmission Time Interval (TTI). The simulation configuration included 21 identical 5G cells that form a compact network, representing a typical deployment scenario in 5G networks:

- 7 BTSs, 3-sector each (21 cells in total),
- Time Division Duplex (TDD),
- Inter-Site Distance: 200m,
- Operating band: 3500 MHz,
- Cell bandwidth: 100 MHz.

Various traffic profiles have been chosen to examine services with varying levels of burstiness (Tab. 8.1). The streaming service exhibits minimal burstiness since traffic is sent according to the codec rate. An File Transfer Protocol (FTP) service with reading times following an exponential distribution is used to represent typical Internet traffic. Lastly, FTP service with zero reading time is utilized to mimic what is known as full buffer traffic, where users wish to download data continuously at the highest possible speeds. Alongside the service types, a specific number of users have been randomly simulated, each generating a specified service demand.

Table 8.1: *Simulation Scenarios*

Scenario ID	Duplex mode	Service	Data Rate/Volume	Reading time [s]	Avg nb of users
1	TDD	Streaming	1.152 Mbps, speech activity 50%	n/a	1, 3, 5, 10, 20, 30
2	TDD	FTP	8.152 MB per call	10 (exponential)	10, 20, 30, 40
3	TDD	FTP	0.721 MB per call	0	1, 10

The simulator generates a table of instantaneous throughput values, along with trajectories per TTI (Fig. 8.2), which are configured to 0.5 milliseconds. According to the analysis, the simulator takes around 30 seconds to stabilize, during which all users initiate their first calls, followed by randomized reading times with an exponential distribution. This stabilization period with the specified TTI results in 60000 samples for further analysis. However, a drawback of such a sophisticated system-level simulator is the substantial computational power and time it demands to produce results; for instance, simulating 30 seconds for 21 cells requires several days (from 2 to 6, which depends on the bandwidth and number of UEs) of computation on a single processor. Alternatively, similar outcomes can be obtained from live network measurements, enabling the method to be applied to other simulated or real network configurations in the future.

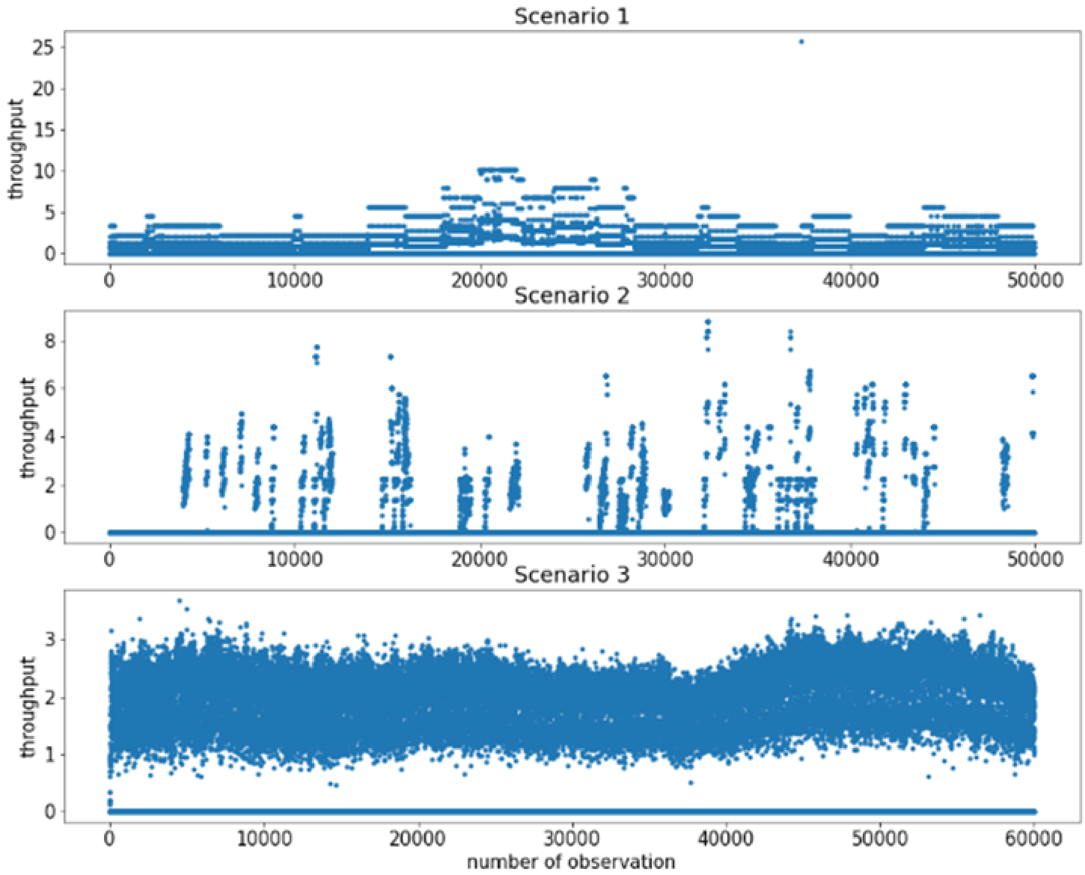


Figure 8.2: *Cell trajectories per scenario from simulator (all cells).*

8.3 Multiplexing Gain Algorithm

8.3.1 Overview

As mentioned in the previous chapter, a critical drawback of the described system-level simulator is its computational complexity and the time required to generate cell trajectories. Consequently, this is a key challenge. Another potential source of such data could be a real live telecommunications network. However, what if we wanted to generate such data cheaply, quickly, and without relying on measurements from a live network? Statistically, we would like to be able to reproduce, generate, while preserving probabilistic properties—cell trajectories similar to those produced by our simulator. One possible solution could involve fitting a reliable statistical model (e.g., a time series) to the data simulated by our simulator. Subsequently, we could generate trajectories from this fitted statistical model. However, in the telecommunications problem under consideration, we have a large number of variables influencing throughput trajectories, such as specific scenarios with varying network setups, traffic loads, and signal propagation conditions. Due to this, the space of possible statistical models that can be reliably fitted and validated seems to be very rich.

The second solution, which was applied in this study, was to apply the bootstrap statistical method. The bootstrap method is a statistical technique that has revolutionized the way researchers approach problems of estimation and statistical inference. Its origin is related to Bradley Efron, an American statistician who, in the 1970s, published a series of papers presenting the basic concepts and applications of this method [101], [102]. Bootstrap is a resampling technique that involves repeatedly sampling with replacement from the original dataset. Technically, sampling with replacement allows some observations to appear multiple times in the new samples, while others may not appear at all. This provides a more representative picture of the data's variability and avoids situations where each sample is identical to the original. In this way, many new artificial samples, called bootstrap samples, are created from the original dataset. These are then used for further statistical and domain-specific analyses. This allows us to obtain probabilistic copies of our original dataset (in our case, throughput trajectories) without making any strong assumptions and without the need for a reliable statistical model fitting process. The main advantage of the bootstrap method is its versatility; it can be applied to a wide range of data. The method is also easy to understand, interpret, and implement. It does not rely on any strong assumptions and works very quickly [103].

The traditional bootstrap method is not suitable for data with dependencies, such as cell trajectories that exhibit a time series with a dependent structure. Therefore, drawing from the literature [104], [105], [106] the focus was on using the block bootstrap approach, which preserves the dependency structure during resampling. This technique involves partitioning the data into blocks to maintain interobservation dependencies, followed by shuffling these blocks to generate random samples. Various methods, including Block Bootstrap, Moving Block Bootstrap, Circular Block Bootstrap and Stationary Bootstrap, leverage block resampling [64]. Among these, the Stationary Bootstrap was selected as it aligns well with the characteristics of the simulation data in this experiment.

8.3.2 Selection of Block Size

In the Stationary Bootstrap method, blocks with exponentially distributed lengths are used, introducing randomness in block sizes while requiring a specified average block size. The key question then arises: How to determine the optimal average block size? The solution algorithm for this challenge is detailed in [107], with an improved version presented in [108]. The ideal average block length depends on the trajectory length and sample autocorrelations. To implement the resampling process, the arch.bootstrap.StationaryBootstrap class from the "arch" package [109] was used.

The theoretical structure of the 5G radio data, called the TDD frame structure, is shown in Fig. 8.3. This frame comprises a total of 40 slots with an equal DL/UL ratio of 4:1. Notably, slots 21 and 22 out of every 160 slots are designated as tracking slots instead of Downlink slots. Consequently, the complete repetitive pattern encompasses 160 slots. As traffic in the Downlink direction has the highest achievable throughputs, it was decided to focus on these slots. Therefore, the SSB, Uplink, tracking, and PRACH slots do not transmit traffic and exibit zero values. The values for the Special slot are lower than those for the Downlink slots as the throughput has to be reduced to limit interference between different slot types. Adjusting the MG algorithm is imperative to maintain the signal characteristic and ensure consistency across all slot types within each trajectory. This necessitates the adoption of a fixed window size that is a multiple of 160. Initially, the MG algorithm computes the best average block length, followed by identifying the nearest multiple of 160. Subsequently, utilizing this best window size, 1000 bootstrap trajectories are simulated for each cell configuration. These trajectories are then categorized into three groups on the basis of their indices: Uplink slots, Downlink slots, and the remaining slots.

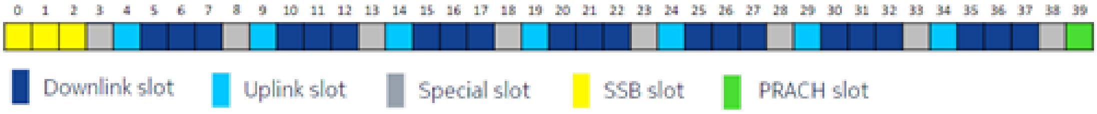


Figure 8.3: *TDD Frame Structure pattern.*

Finally, data were simulated for 21 cells, each configuration serving as input. A random selection of 200 replacement indices was made from the 21 available cells and the best average block size was computed for each cell. Subsequently, data for 200 cells, mimicking the distribution and properties of the initial cells, were generated using a Bootstrap-based generator (Fig. 8.4).

8.3.3 Validation

Initially, a visual inspection was performed to ensure the functionality of the MG algorithm. In Fig. 8.5, an illustration was shown comparing a typical normalized original trajectory with a bootstrap trajectory produced by the MG algorithm. It is evident that the pattern was accurately replicated and the relationship between the data points is apparent, although there are discrepancies between the two trajectories, which aligns with the initial expectations for the computational study. Subsequently, the results obtained were analytically validated using quantile lines.

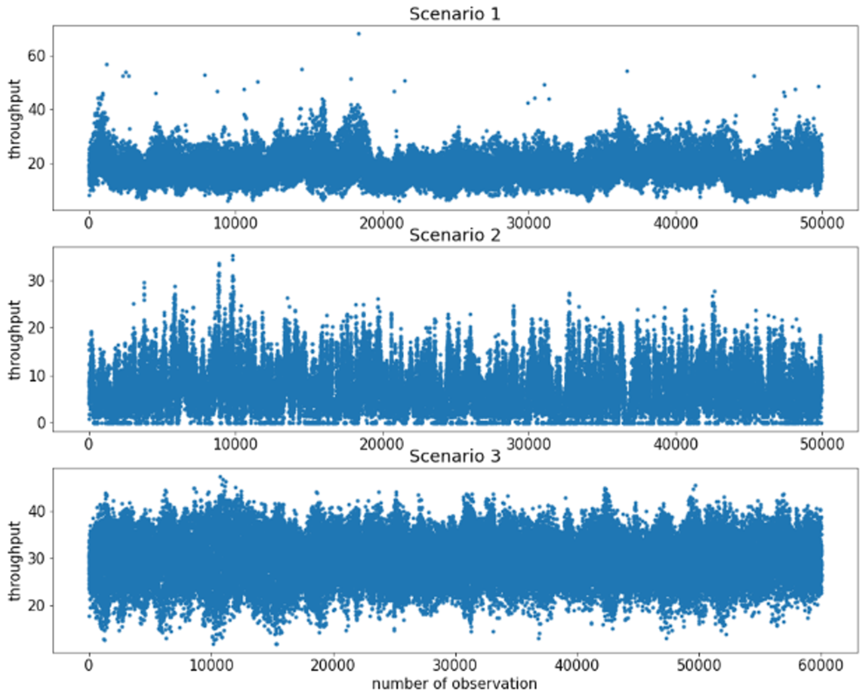


Figure 8.4: *Sample momentary throughput from bootstrap simulations.*

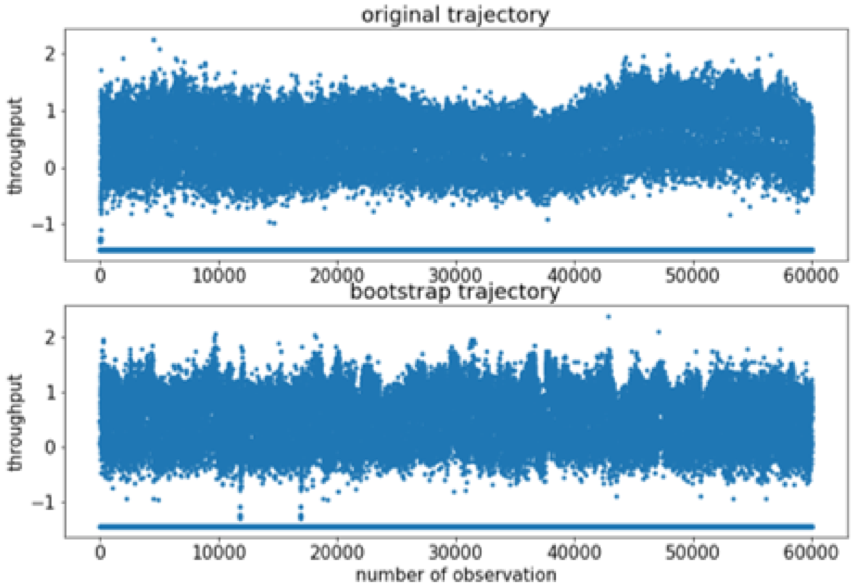


Figure 8.5: *Comparison of normalized original and bootstrap trajectories.*

Quantile lines $Q_{(0.05)}$ and $Q_{(0.95)}$ are calculated separately for the uplink and downlink slots. The illustrative results are presented in Figs. 8.6 and 8.7. It is evident that only a small number of individual slots deviate from the quantile lines, typically accounting for approximately 5% of slots in all cells after numerical verification.

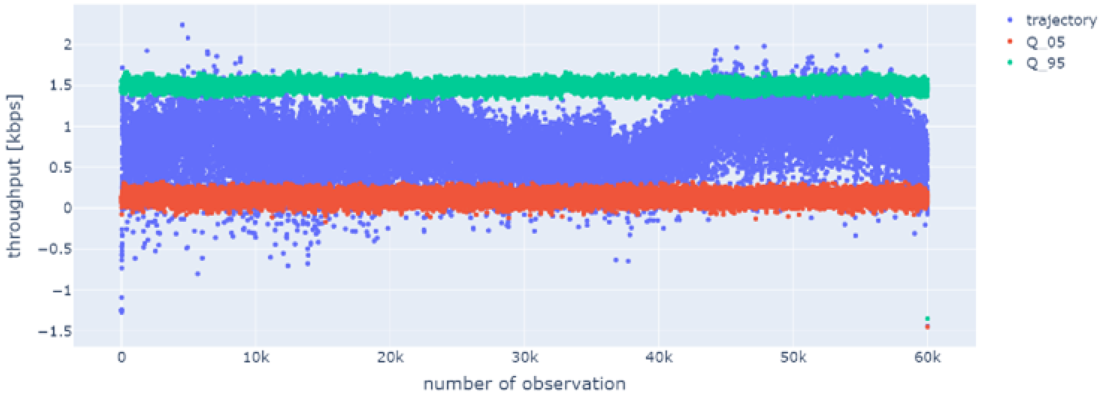


Figure 8.6: *Quantile lines and sample trajectories for Downlink slots.*

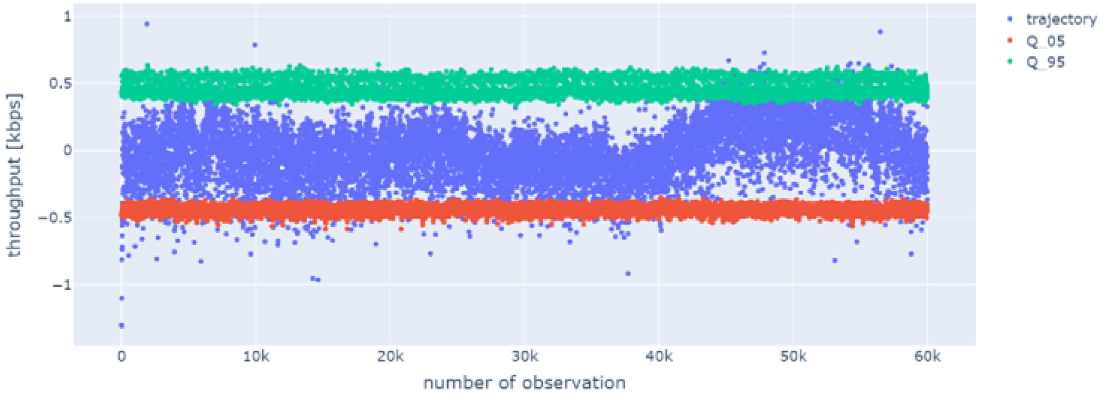


Figure 8.7: *Quantile lines and sample trajectories for Special slots.*

8.4 Results Analysis

8.4.1 Traffic Type Impact on Multiplexing Gain

Fig. 8.8 illustrates the dependency of MG to number of aggregated cells for three scenarios. The most significant improvements are observed for Scenario 2, then Scenario 1 and the least improvement (but still significant) can be observed for Scenario 3. The reasoning behind such behavior is related to the type of service and the characteristics of the traffic. Scenario 2 was configured with bursty FTP traffic, where UEs request data transmission with random reading time (Tab. 8.1. After each request, depending on the radio link quality and other data tranmission in the same cell, UE often gets high throughput rates and the data are transmitted over a short period of time. This short transmission times with high rates decrease the probability that any other traffic will be transmitted in the same slots, which eventually increases the MG. On the other hand, Scenarios 1 and 3 were configured with static traffic, that is, streaming with constant data rate and FTP with zero reading time (meaning that new data transmission requests are generated just after received last data from previous request), respectively.

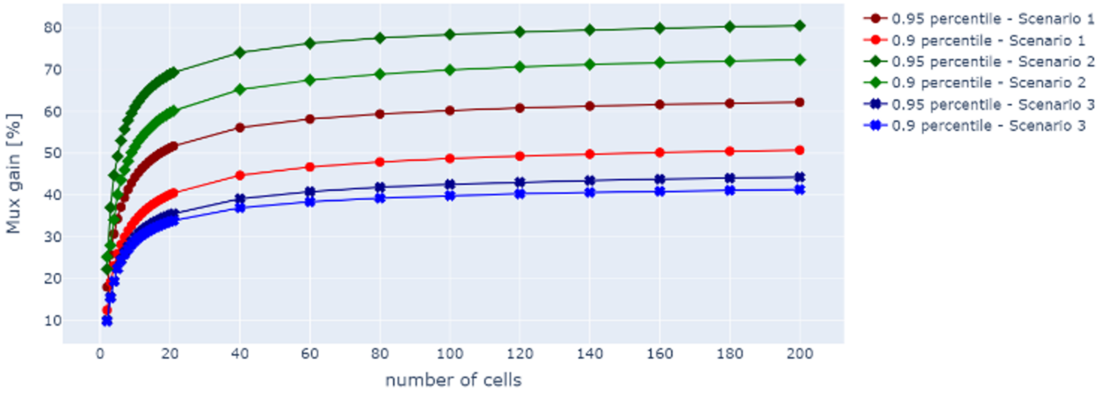


Figure 8.8: *Multiplexing Gain as a function of aggregated cells number for three scenarios.*

8.4.2 Number of Users Impact on Multiplexing Gain

The quantity of users (UEs) is another factor that influences the MG level. As the number of users increases alongside the consistent traffic assumptions per user, the total traffic within each cell also increases. Consequently, as traffic escalates, the MG level diminishes due to reduced multiplexing capacity. This trend is observable in all scenarios, such as those illustrated in Fig. 8.9 and Fig. 8.10.

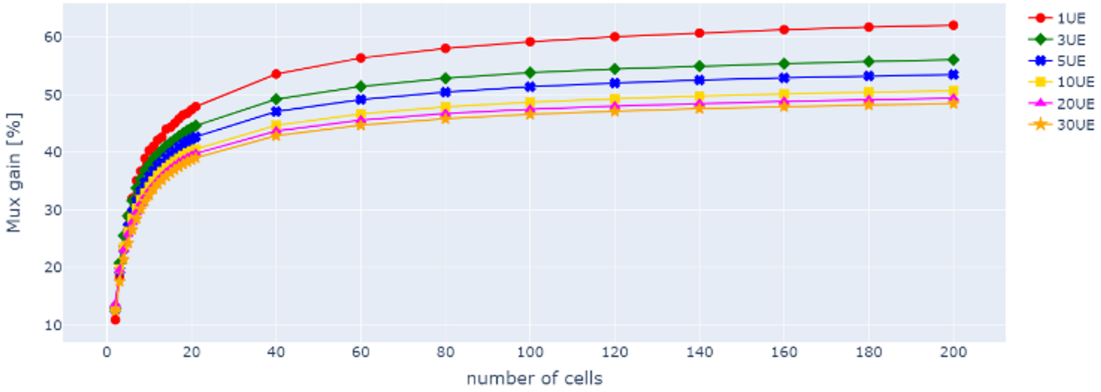


Figure 8.9: *Multiplexing Gain as a function of aggregated cells number for Scenario 1 and P=0.9.*

8.4.3 Percentile Impact on Multiplexing Gain

The choice of link capacity based on the percentile of throughput achieved on the link also affects the MG. As the percentile increases, the MG also increases. Fig. 8.11

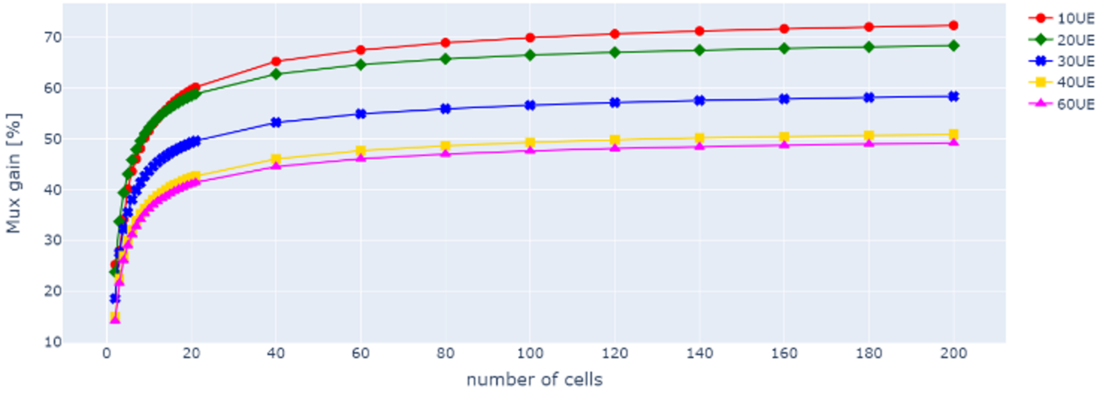


Figure 8.10: *Multiplexing Gain as a function of aggregated cells number for Scenario 2 and P=0.9.*

illustrates that the MG grows linearly from P=0.7 to P=0.9. When P=1, the capacity reaches the peak rate, which could be significantly higher than the other, more typical, throughput values. Consequently, the findings indicate that the MG at P=1 deviates from the linear trend and stands out from the lower percentiles.

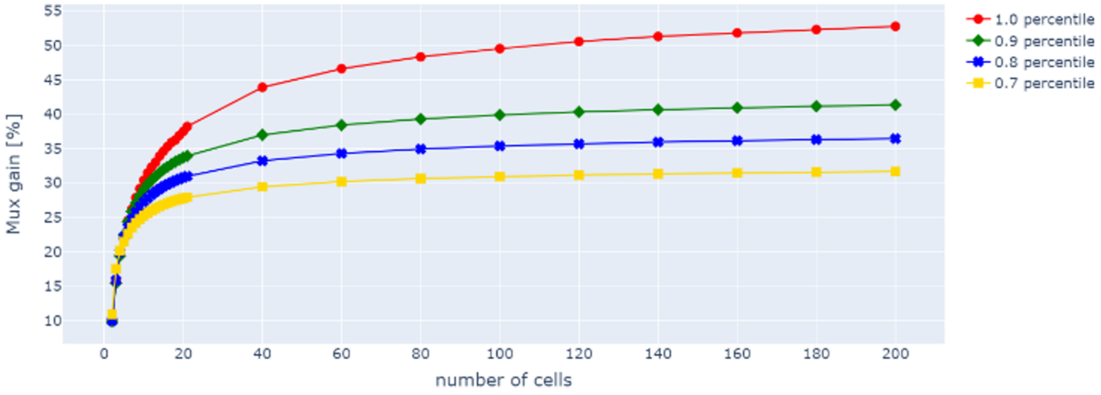


Figure 8.11: *Multiplexing Gain as a function of aggregated cells number for Scenario 3 and UE=10.*

8.4.4 Number of Cells Impact on Multiplexing Gain

From the data presented in this section (Figs. 8.8 - 8.11), it is evident that the MG increases as more cells are aggregated. The growth function is not linear, with the most significant gains observed when aggregating a small number of cells, typically between 2 and 20. Although adding another cell at any stage results in additional MG, the incremental gain decreases compared to aggregating multiple cells due to traffic flow saturation. Once a certain volume of traffic is aggregated, there is limited capacity to accommodate additional traffic. Consequently, the subsequent increase in traffic leads to minimal statistical gains. This saturation effect is similar to a "knee point" indicating the number of cells that contribute the majority of the MG. Factors such as the number of users, traffic patterns and the selected percentile influence the MG, causing the "knee point" to shift. However, when examining the horizontal axis that denotes the number of cells, this critical point typically falls within the range of 20-40 cells. Understanding this behavior is essential for effective transport network planning. From a capacity standpoint, it is more advantageous to position aggregation points for every 20-40 cells rather than consolidating a larger number of cells.

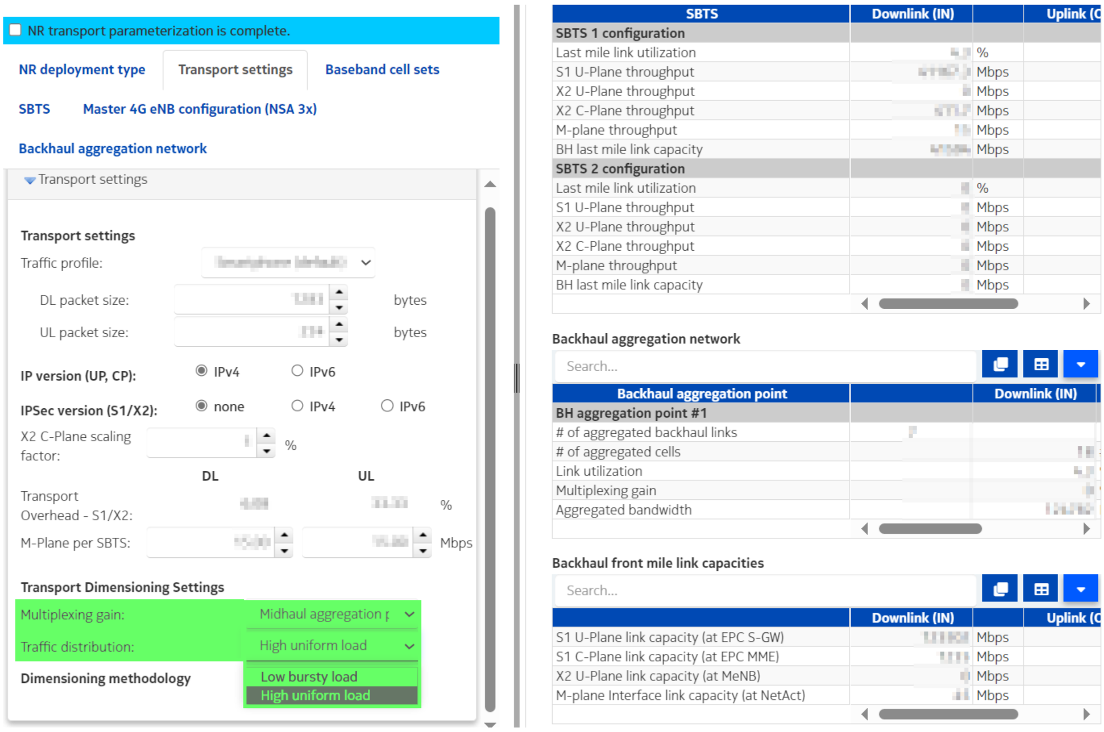
8.5 Industrial Applications

Dimensioning and planning processes that are established in Nokia are supported by a proprietary tool developed by the Network and Performance Engineering department, the host of the author of this dissertation. This tool, with 4G origins, is able to estimate required BTS baseband and link capacities. Furthermore, it is used for midhaul and backhaul aggregation point capacity estimation. Therefore, once the MG algorithm was created and validated, it has been developed in the tool to improve the precision of aggregation point link capacity estimation. To simplify tool usage (transport dimensioning settings highlighted with green color in Fig. 8.12), possible scenarios have been limited to "Low bursty load" and "High uniform load" (Scenario ID 2 and 3 from Tab. 8.1), which are most often required for, respectively, new 5G deployments and existing high traffic deployments.

The industrial application of the methods, algorithms and results in commercial tools and processes is highly important in the context of this doctoral dissertation being developed in the industrial doctorate program. Therefore, two further case studies described in the following sections present validation and usage of these results as a part of dimensioning process in Nokia.

8.5.1 Case Study - Microwave Link Capacity

For microwave transmission design, which is often used to interconnect radio access network, it is typically necessary to select link capacity for a location with multiple base stations collocated (alternatively, for multi-RAT BTS with, e.g., several 4G/5G frequency layers) - this results in the presence of aggregation points that comprise 3-18 cells. In the dimensioning process, the transport capacity for each individual cell is calculated first. Second, aggregated capacity is estimated according to the MG algorithm with two "extreme" values considered depending on the required trade-off between bandwidth savings and delay/congestion probability: low bursty and high uniform load.



☐ NR transport parameterization is complete.

NR deployment type | Transport settings | Baseband cell sets

SBTS | Master 4G eNB configuration (NSA 3x)

Backhaul aggregation network

Transport settings

Transport settings

Traffic profile:

DL packet size: bytes

UL packet size: bytes

IP version (UP, CP): ● IPv4 ○ IPv6

IPSec version (S1/X2): ● none ○ IPv4 ○ IPv6

X2 C-Plane scaling factor: %

	DL	UL	
Transport Overhead - S1/X2:			%
M-Plane per SBTS:			Mbps

Transport Dimensioning Settings

Multiplexing gain: Midhaul aggregation p

Traffic distribution: High uniform load

Dimensioning methodology

Low bursty load

High uniform load

SBTS	Downlink (IN)	Uplink (O
SBTS 1 configuration		
Last mile link utilization	%	
S1 U-Plane throughput	Mbps	
X2 U-Plane throughput	Mbps	
X2 C-Plane throughput	Mbps	
M-plane throughput	Mbps	
BH last mile link capacity	Mbps	
SBTS 2 configuration		
Last mile link utilization	%	
S1 U-Plane throughput	Mbps	
X2 U-Plane throughput	Mbps	
X2 C-Plane throughput	Mbps	
M-plane throughput	Mbps	
BH last mile link capacity	Mbps	

Backhaul aggregation network

Search...

Backhaul aggregation point	Downlink (IN)
BH aggregation point #1	
# of aggregated backhaul links	
# of aggregated cells	
Link utilization	
Multiplexing gain	
Aggregated bandwidth	

Backhaul front mile link capacities

Search...

	Downlink (IN)	Uplink (O
S1 U-Plane link capacity (at EPC S-GW)	Mbps	
S1 C-Plane link capacity (at EPC MME)	Mbps	
X2 U-Plane link capacity (at MeNB)	Mbps	
M-plane Interface link capacity (at NetAct)	Mbps	

Figure 8.12: *Dimensioning tool implementing Multiplexing Gain algorithm.*

The analysis of results for two network Operators is presented below. Each configuration of a site consisting of one to many BTSs with multiple cells is given in Tab. 8.2. Each such site is connected by microwave link to the network. Thanks to the link measurements, each momentary value has been collected, whereas the peak values are given in the Tab. 8.3. Finally, the capacity for every site configuration has been calculated using the MG algorithm.

Table 8.2: *Site configurations for Operator 1.*

RAN technology	4G	4G	4G	4G	4G	5G	5G
Band	L700	L800	L1800	L1800	L2100	2600	3500
Spectrum [MHz]	5	15	15	20	15	100	2x100
No cells per BTS	3	3	3	3	3	3	3
Config 1	x	x	x	x	x		x
Config 2		x	x	x	x	x	
Config 3		x	x	x			x
Config 4		x	x			x	

For configuration 4, the peak throughput value is between the two estimated values, which shows a good fit of the MG algorithm to the real data. This means that the traffic

Table 8.3: *Measured throughput and estimated aggregated DL link capacity per site configurations for Operator 1.*

[Gbps]	Config 1	Config 2	Config 3	Config 4
Max measured throughput	0.977	0.868	1.566	1.332
Estimated capacity with MG algorithm for high uniform load	1.362	0.763	1.390	1.588
Estimated capacity with MG algorithm for low bursty load	1.041	0.588	1.084	1.257

transmitted over these sites is moderate (between low and high). For configuration 1, the peak value is below the estimated capacity for low bursty load, which shows that the traffic over this site is small (this is the smallest site of the exercise with only three cells). For configurations 2 and 3, the peak value exceeds the estimated capacity for high load. This behavior has been further investigated. Figs. 8.13 and 8.14 present a detailed analysis of the measured throughput values with its distribution.

The conclusion is that the maximum values observed in the network for these two site configurations were observed only once over four months. Such isolated traffic surges are effectively managed by transport QoS mechanisms that restrict instantaneous throughput and queue traffic. This limitation is contingent on the dimensioning and Committed Info Rate configurations. From a service point of view, it can be entirely transparent. Thus, the method is also effective for these two configurations (2 and 3), indicating that the MG algorithm can, in general, offer a reasonable estimation of the necessary capacity.

Configurations deployed in Operator 2 network are given in Tab. 8.4. The measured throughput and estimated aggregated DL link capacity per site configurations are given in Tab. 8.5. In this case the same analysis has been done as for Operator 1 and it shows that the MG algorithm estimates well required capacity, the peak throughput values are below estimated capacity.

Table 8.4: *Site configurations for Operator 2.*

RAN technology	4G	4G	4G	4G	5G	5G
Band	L800	L1800	L2100	L2600	L2600	3500
Spectrum [MHz]	10	20	20	20	80	100
No cells per BTS	3	3	3	3	3	3
Config 1	x	x	x	x	x	
Config 2	x	x	x	x	x	x

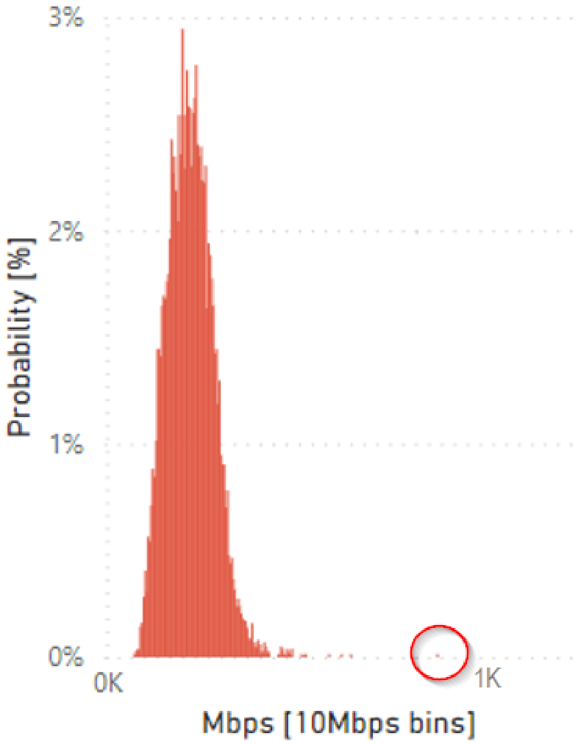


Figure 8.13: *Throughput probability distribution of Operator 1's microwave link for configuration 2. Red circle marks the maximal value.*

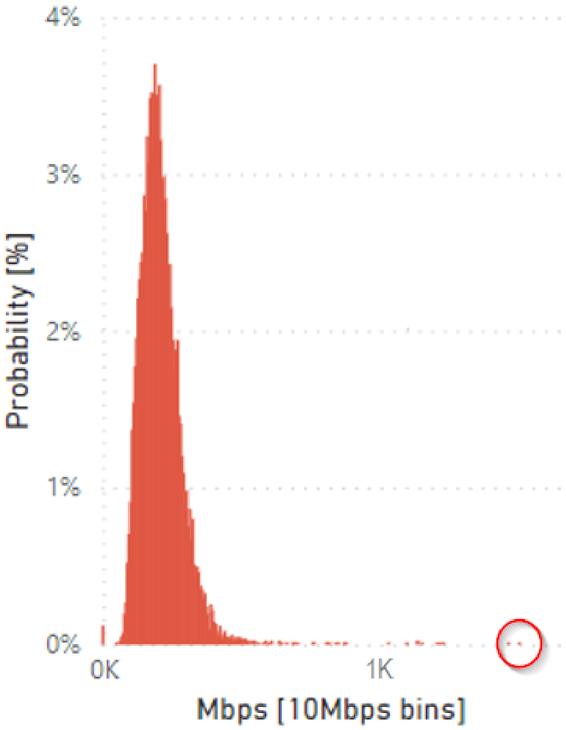


Figure 8.14: *Throughput probability distribution of Operator 1's microwave link for configuration 3. Red circle marks the maximal value.*

Table 8.5: *Measured throughput and estimated aggregated DL link capacity per site configurations for Operator 2.*

[Gbps]	Config 1	Config 4
Max measured throughput	0.652	0.888
Estimated capacity with MG for high uniform load	0.804	1.606
Estimated capacity with MG for low bursty load	0.619	1.227

8.5.2 Case Study - Cloud BTS Transport Capacity

As outlined in Sec. 1.3.4, the shift to cloud-based BTS architecture complicates the process of dimensioning the capacities of the transport links. Fig. 8.15 presents a recent case study that illustrates the transport topology used to connect DU with CU. The backbone network links the DUs and the core network with a site solution for the CU, which includes two routers and two switches. The objective was to estimate the link capacity for network points marked with arrows and capital letters.

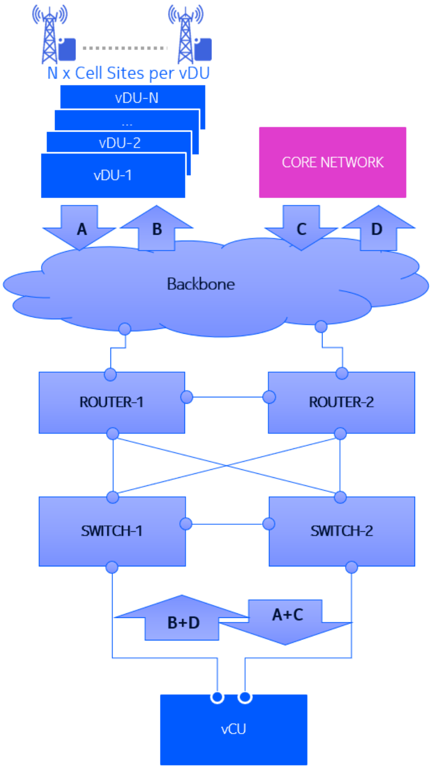


Figure 8.15: *Cloud BTS architecture from real case study.*

The study findings are presented in Table 8.6. Two scenarios are analyzed: one with and one without MG estimation. The variation between these estimations spans from 17% to 34%, indicating a significant decrease in link capacity with consideration of statistical MG. Moreover, these results prove the need to include this factor in the dimensioning process.

Table 8.6: *Estimated link capacities for cloud BTS case study.*

Link	No multiplexing gain [Gbps]	With multiplexing gain [Gbps]
A - vDU to vCU	0.921	0.762
B - vCU to vDU	0.168	0.140
D - vCU to Core	0.945	0.618
C - Core to vCU	0.175	0.114
A+C - vCU Ingress	1.114	0.885
B+D - vCU Egress	1.114	0.787

8.6 Summary

This chapter presents an original technique, using system-level traffic data, for estimating statistical MG of aggregated 5G transport links. The MG algorithm can use actual data from a simulator or live network, improving its practical applicability and readiness for integration into RAN planning and dimensioning systems for commercial use. The key benefit of this proposed approach lies in its utilization of real-world measurements, focusing on actual radio performance rather than theoretical values. The proposed MG algorithm enables the scalability of the simulation outcomes, extending from a 21-cell network to 200 or more cells. Moreover, the MG algorithm does not rely on a specific model for dimensioning BTS transport interface capacity, serving as a complementary tool to enhance existing aggregation point methods. In this context, it enhances the method for forecasting throughput and delay for each NS. Different slices may utilize varied transport paths, which must be taken into account during the planning and dimensioning of the transport link. The total capacity of individual transport links can consist of the same type or different types of slices. Nonetheless, the transport planning MG algorithm can be used to accurately estimate the aggregated capacity.

As a verification, a comparison was made between the trajectories produced by the bootstrap method (Fig. 8.4) and the simulated data using quantile lines (Fig. 8.6 and Fig. 8.7). The analysis indicated a good fit, with 90% of the generated data samples falling within the quantile lines 5% and 95% of the original signal (simulated).

The results indicate that the MG increases as the number of combined cell traffic flows over a single midhaul link rises. This increase follows a logarithmic pattern and the "knee point" varies depending on factors such as gNB setup, traffic characteristics, and selected percentile. In scenarios with high traffic and peak loads, the identified "knee point" usually falls between 20 and 40 cells, resulting in gains of 25-45% [68]. In summary, the best placement for an aggregation point between DU and CU is where it can consolidate traffic from 20-40 cells. Going beyond this range is unlikely to produce significant changes in the practical MG. In addition, the proposed approach reduces the computational time from days to seconds, which is crucial for network planning recommendations and ultimately improves the efficiency and flexibility of services provided to telecommunication Operators.

Finally, the MG algorithm was integrated into a professional Nokia tool and applied in various case studies. Two of these studies have been presented, demonstrating the alignment of the estimations with measured values from microwave links in mobile networks and highlighting its relevance to cloud BTS dimensioning.

Chapter 9

5G Network Slicing Dimensioning Framework

The research covered in this doctoral dissertation thus far complements one another, forming a comprehensive 5G Network Slicing Dimensioning Framework, referred to in this chapter simply as the framework. All verified, tested, and developed methods, models, and procedures are integrated.

9.1 Description of Framework Elements

In light of emerging ideas, a framework for forecasting and dimensioning is proposed (Fig. 9.1), which is a data-centric model that incorporates the concept of the DT as outlined in [70]. Essentially, the proposed framework can function autonomously as a forecasting and scenario analysis tool applicable to (sliced) network dimensioning and traffic management, or it can operate as a fundamental element of a versatile DT. This structure has been modularly designed to facilitate easy validation, administration, and adaptation to particular scenarios.

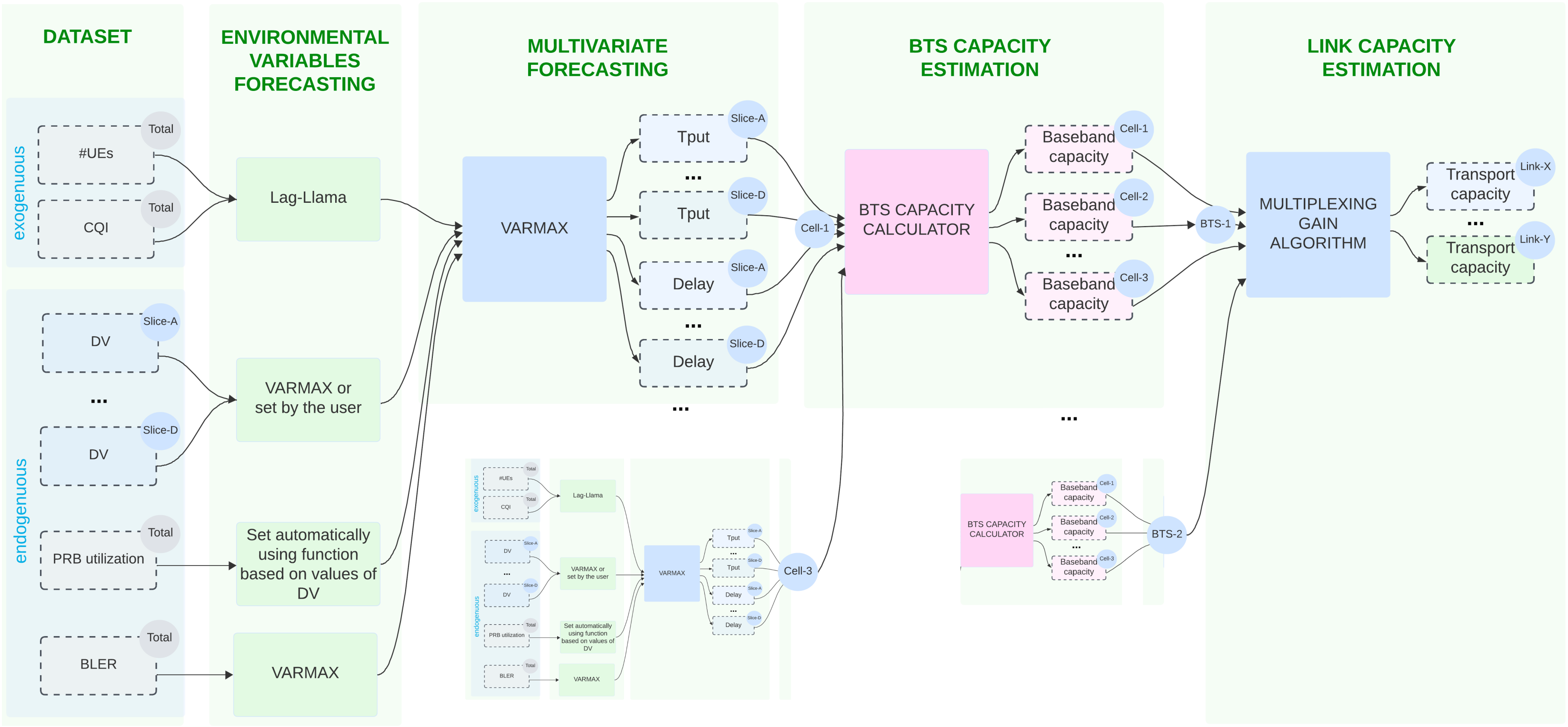


Figure 9.1: *5G Network Slicing dimensioning framework.*

It is important to note that conducting DT-based simulations necessitates linking with the data platform [110] in order to update the model when the relationship between inputs and outputs changes (such as configuration adjustments or software upgrades) or differs (such as configurations not present in the training data). Hence, consideration must be taken to ensure the continuity of the link between physical and digital networks. The intention is to integrate this process into a continuous delivery cluster similar to that outlined in [73].

9.1.1 Dataset Module

The dimensioning process involves collecting input data that detail the network's operational conditions and the services it is anticipated to offer. In this study, a dataset comprising hourly averaged time series data from 5G BTS in a live network deployment has been used (Sec. 3.1). Data have been collected at both the cell and network slice levels, facilitating the forecasting and subsequent dimensioning of capacity for each slice and the entire cell.

Furthermore, to investigate the QoS demands (instantaneous data rates) of the wireless interface and the evolving traffic profiles of multiple cells simultaneously, a specialized Nokia system-level simulator has been utilized. This comprehensive dynamic system-level simulator is designed to emulate user plane performance, encompassing throughputs and delays, using the provided radio resource management features and algorithm options (Sec. 8.2).

Using real network data and occasionally system-level simulations, the framework can address the issue of *a priori* defined TM discussed in Sec. 1.3.2.

9.1.2 Environmental Variables Forecasting Module

The variables chosen for the input of a multidimensional delay and throughput forecasting model (Tab. 3.1) require preprocessing, and for exogenous variables, forecasting is also needed. Specifically, forecasting #UEs and CQI using a one-dimensional model is essential as input for multidimensional forecasting models.

In Chapter 5, several one-dimensional models, namely SARIMA, Prophet, Chronos, and Lag-Llama, were examined and compared to accurately predict #UEs and CQI (individually). Among the validated models, both Prophet and Lag-Llama delivered the most precise results, which were generally comparable. Ultimately, Lag-Llama was selected because of its extensive potential for fine-tuning, as it can utilize more data for training, which will be thoroughly investigated in future research.

For endogenous variables, the following approach has been taken. DV per slice and BLER are forecasted by VARMAX, although DV can be manipulated by the framework user to simulate "what-if" scenarios (Sec. 9.2). PRB utilization is changed automatically using the function based on the DV values, as both features show a high correlation in the research (Sec. 3.4).

To establish the initial state for endogenous variables, the following method is used. When there is no change to the DV at time $T-1$, the endogenous variables forecast at time T is performed using the VARMAX model, and the initial state is simply the complete dataset at time $T-1$. This can be considered the initial reference state. If the user wishes to simulate a scenario where the DV for a specific slice changes by $x\%$, the initial state is set as $(100+x)\% \cdot y$, where y is the forecasted value of DV at time T (the reference state). For BLER, the initial state is equal to its reference state.

Once the initial values of DV are computed, the PRB utilization can be determined. It has been noted that PRB utilization can be expressed as a function of data volumes. Several methods for forecasting PRB utilization have been explored, such as forecasting using functional dependencies (polynomials, exponential functions, logarithms) or random forest models, multiple linear regression, ridge, and lasso (Figs. 9.2 - 9.4). For further information on these models, see [89].

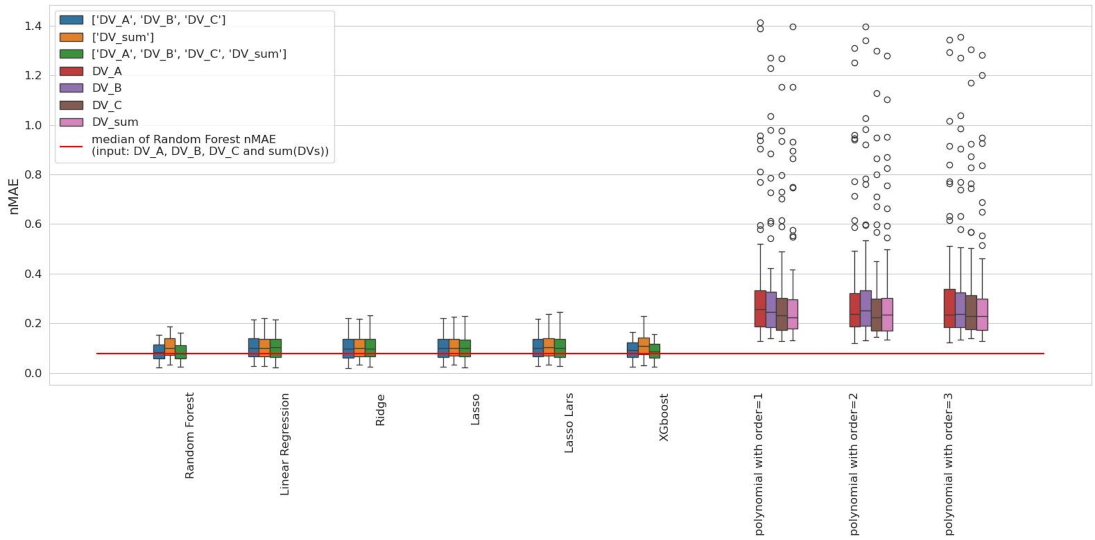


Figure 9.2: *The comparison of nMAE for PRB utilization short-term forecasting.*

The selected methods are straightforward models that are computationally efficient. KPIs are interrelated, and forecasting the dependent variable is not a complex task (hence a simple solution is preferred). The best results were achieved using a random forest, which takes as input the DV in each Slice, BLER, CQI, and #UEs.

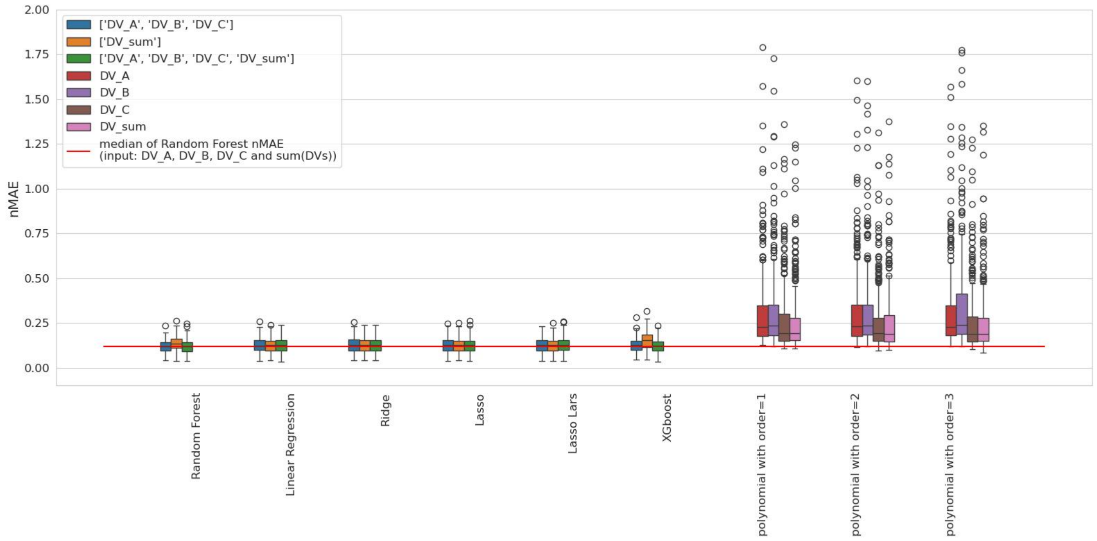


Figure 9.3: *The comparison of nMAE for PRB utilization long-term forecasting.*

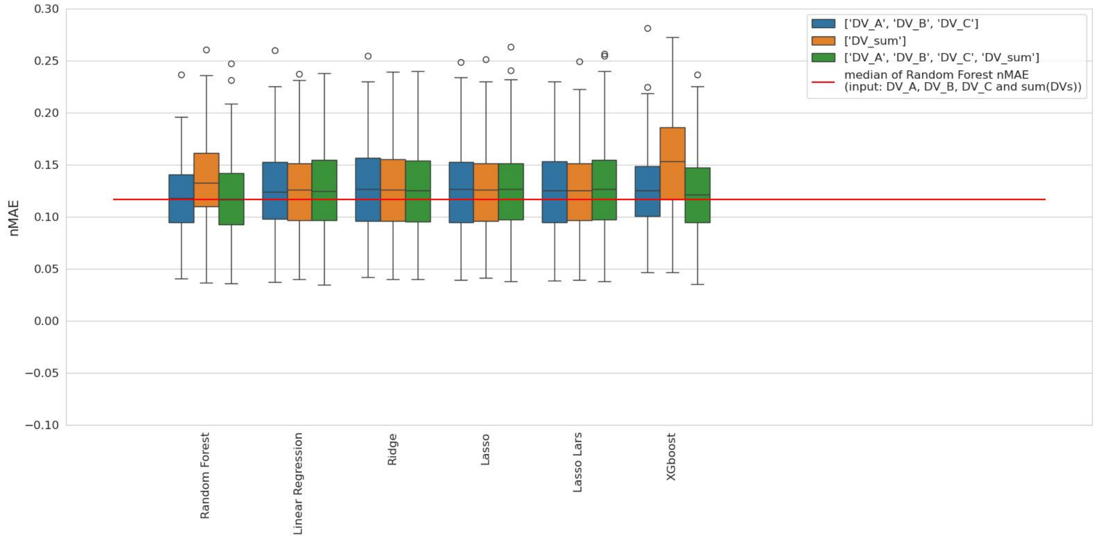


Figure 9.4: *The comparison of nMAE for PRB utilization long-term forecasting (zoom on the lowest values).*

9.1.3 Multivariate Forecasting Module

The core of this module revolves around the forecasting model, which is trained using actual traffic and environmental data. This comprehensive model forecasts both throughput and delay at a detailed cell and slice level. When applied in the dimensioning process, it eliminates the need for a predefined TM (Sec. 1.3.2). Various techniques have been tested, and VARMAX has shown the best and most accurate forecasting results. By utilizing a pre-trained model, users can conduct "what-if" scenario simulations. By adjusting historical model inputs based on scenario requirements, future estimates can be made for slice throughput and delay. Given that some input features are interrelated (as discussed in Section 3.4), any modifications to the inputs must take into account their connections (as discussed in the previous section).

A multidimensional model was created to predict throughput and delay based on environmental KPIs because it is necessary to investigate the relationship between DV and the various factors involved in modeling throughput and delay. If DV increases in the simulations, other environmental variables must also be taken into account, as changes in DV would affect them. It is inaccurate to assume that DV changes alone, such as without changes in PRB utilization, for a realistic simulation of a 5G network. The model takes past values of endogenous variables (including throughput and delay) and current values of exogenous variables as input (Fig. 9.1). Thus, forecasting requires knowing the future values of exogenous variables. Therefore, one-dimensional models are devised to forecast these variables (in the Environmental Variables Forecasting Module), and their forecasting results are then utilized in the main model.

In this section, throughput and delay metrics are predicted for each configured slice within a cell. Consequently, to dimension at the BTS level, this procedure must be repeated for all cells associated with this BTS.

9.1.4 BTS Capacity Estimation Module

Utilizing actual data from pre-trained cellular models in the dimensioning process results in precise traffic forecasting and facilitates capacity estimation for future network development. Upon assuming that service usage will continue to increase at the current rate, the model can be used to forecast future capacity. To determine the necessary capacity and infrastructure, throughput and delay are forecasted for each slice.

Subsequently, for each cell configuration and slice, it is assessed when the slice capacity or slice QoS delay requirements will be met. Identifying the time when capacity limits will be reached, proactive system configuration adjustments can be made to enhance slice capacity. This method can also be applied to long-term dimensioning (planned for future research), where the forecasted time to reach the delay limit will indicate when network expansion is required, and the forecast capacity at the end will indicate the extent of infrastructure expansion needed. By incorporating a "what-if" analysis, it becomes possible to conduct capacity dimensioning for simulated scenarios that consider various potential changes in the evolution of network traffic demand. For example, there could be a significant increase in demand for a specific slice due to a new planned offering from the CSP.

In this module, a crucial component involves calculating the cell and BTS capacity, which is related to understanding the capabilities and limitations of the product. Nokia has already created a database containing such information and a calculator to choose suitable product versions based on QoS requirements. This solution can determine the number of hardware or cloud resources needed. This is the only part of the framework that was not developed in the research associated with this doctoral dissertation. Furthermore, the results of this step cannot be publicly shared as the product limitations are company-confidential.

9.1.5 Link Capacity Estimation Module

An element of the RAN planning process is capacity dimensioning, which seeks to determine the resources necessary to provide services to particular traffic volumes while preserving QoS as outlined in the previous subsection. Within the realm of transport network dimensioning, this primarily entails estimating the capacities of links. This step is generally performed at the BTS level, since traditionally, each BTS had a single link connection to the mobile core network. In contrast, in the case of Cloud-RAN, each RU connects to the DU and later to the CU via the transport network (as detailed in Chapter 8). Traffic from multiple cells is consolidated at some point (with the location of this point depending on the specific topology planning process) within the transport network. In the link capacity estimation module, an algorithm is used to evaluate the statistical transport MG to determine the final link capacity.

9.2 Scenario Simulations with the Framework

This framework can be used for conducting "what-if" analyzes, e.g., for scenarios involving substantial traffic growth, such as when a CSP intends to enhance the usage of specific services or implement new service offerings. By adjusting the amount of traffic in the input while keeping environmental conditions changing according to the forecasts, we can make valid statements regarding the throughput and delay processes observed in actual systems. Fig. 9.5, presents this concept. The framework can be used to forecast when the capacity limit will be reached without any changes to the characteristics of environmental conditions - Scenario 1. It can also be used to predict what will happen once the conditions will change, e.g. the expected DV will increase or decrease as shown in Scenario 2 (Slice A traffic increased by 10%) and Scenario 3 (Slice B traffic decreased by 5%). This shows the concept of how the framework can effectively elucidate and quantify these phenomena through data-driven simulations of sliced wireless networks.

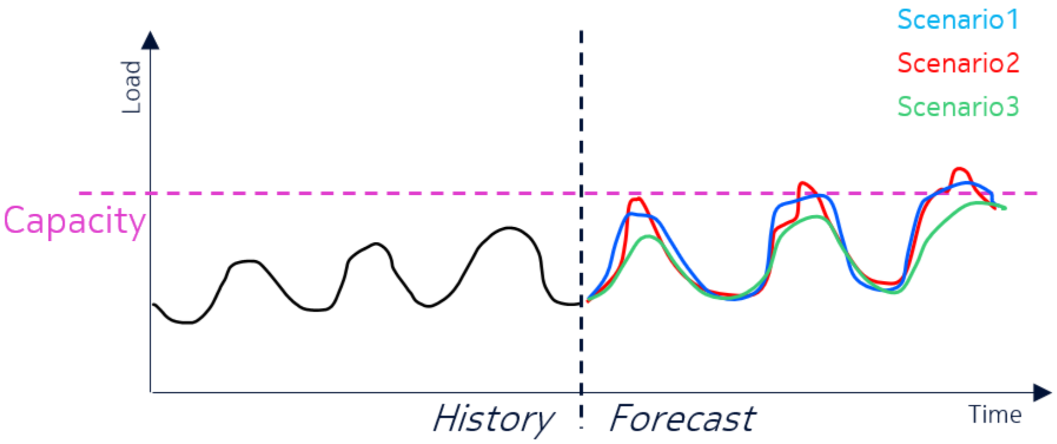


Figure 9.5: *The concept of simulated scenarios (with artificial data).*

In general, this framework represents an initial stage in implementing the DT idea for communication networks. As defined in the literature [111], DT serves as a digital model of a cell of the actual 5G base station. By training the DT using authentic data collected from individual cells, unique twins are generated, adapted to tasks such as forecasting cell traffic and delays, as well as performing hypothetical scenarios. Furthermore, it can support optimization systems to suggest capacity expansions or configure parameters for slice planning.

9.2.1 Evaluation with Real Data

To verify the validity of the simulator and evaluate its applicability and ability to maintain the physical context, experiments were performed on the actual data. Initially, the moments when the DV increased were identified from the data. Two types of changes are observed during these moments: a single peak and a change in the DV, after which the DV remains elevated. Both types were marked accordingly. An illustrative example of a single jump in DV-B is shown in Fig. 9.6.

The complete procedure is as follows:

1. fit model for data before DV change (blue part),
2. set initial state for simulation by taking true values of data volumes for each slice and PRB utilization (red part). The input values for BLER are forecasted by the standard procedure described in Sec. 4.2.1,
3. forecast next 24 hours (green part).

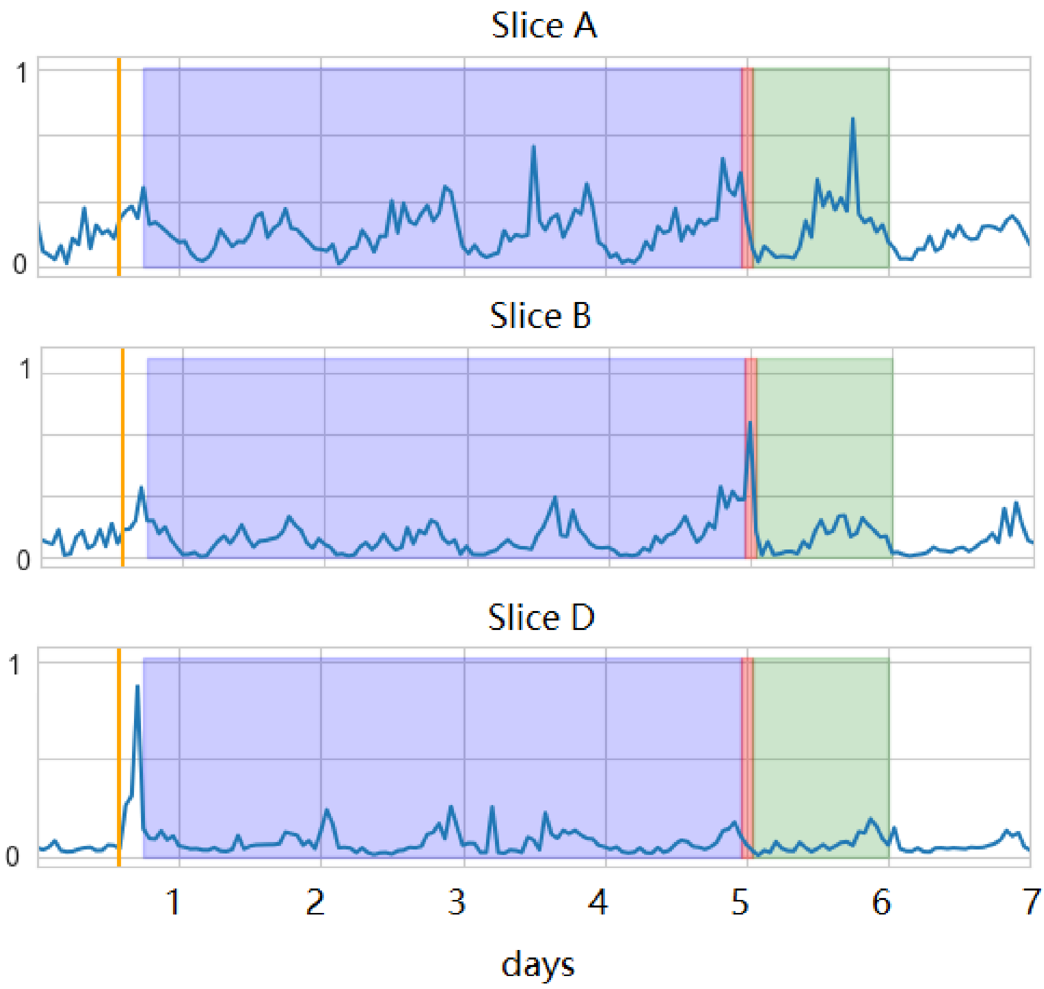


Figure 9.6: *The example trajectories for data volumes across each network slice.*

The simulations utilize the VARMAX(1,0) model (Chapter 6).

Details regarding the actual PRB utilization value are provided in the initial state. This is because alterations in DV lead to corresponding changes in the use of radio resources. A simulation considering only DV changes would not accurately represent reality. An example forecast based on real data is shown in Fig. 9.7. In this scenario: DV_A increases by 267.2%, DV_B increases by 7.98%, DV_D increases by 13.7%, and PRB utilization increases by 80.75%. It should be noted that a large increase in DV in Slice A is associated with a significant increase in PRB utilization.

Table 9.1 shows the information criteria and evaluation statistics for a sample from Fig. 9.7. It is evident that the errors are minimal and the R^2 value is high.

To evaluate the results of all samples, two metrics are utilized: MAE and the RdR score as presented in [112]. The RdR score is a standardized metric derived from Dynamic Time Warping (DTW) and RMSE, showing whether the forecast of the evaluated model

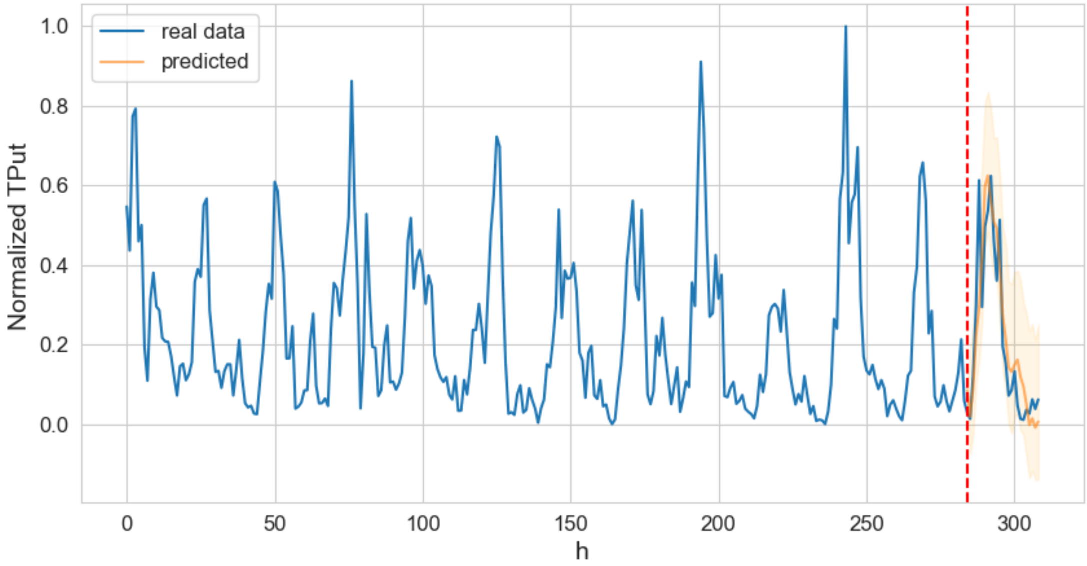


Figure 9.7: *Predicted throughput for Slice A employing VARMAX.*

Table 9.1: *The information criteria values and evaluation metrics for a sample depicted in Fig. 9.7.*

AIC	BIC	HQIC	MAE	RMSE	R2
-6001.57	-5195.14	-5678.26	4.13	5.25	0.77

exceeds the results of other methods. With slight modifications, an alternative version of the RdR score was considered, which evaluates whether the model outperforms its univariate equivalent (ARMA for seasonally decomposed data). In general, the revised RdR score can be represented as follows:

$$RdR_{score} = \frac{RMSE_{score} + DTW_{score}}{2}, \tag{9.1}$$

where

$$RMSE_{score} = 1 - normalized(RMSE),$$
$$DTW_{score} = 1 - normalized(DTW).$$

The normalized RMSE is calculated by first determining the RMSE of the model, then subtracting the minimum reference value, and finally dividing this result by the reference range. Reference boundaries are defined by the RMSE values of an ARMA. Similarly, the normalized DTW is defined in the same way. The MAE is normalized by dividing it by the range to ensure comparability between all samples.

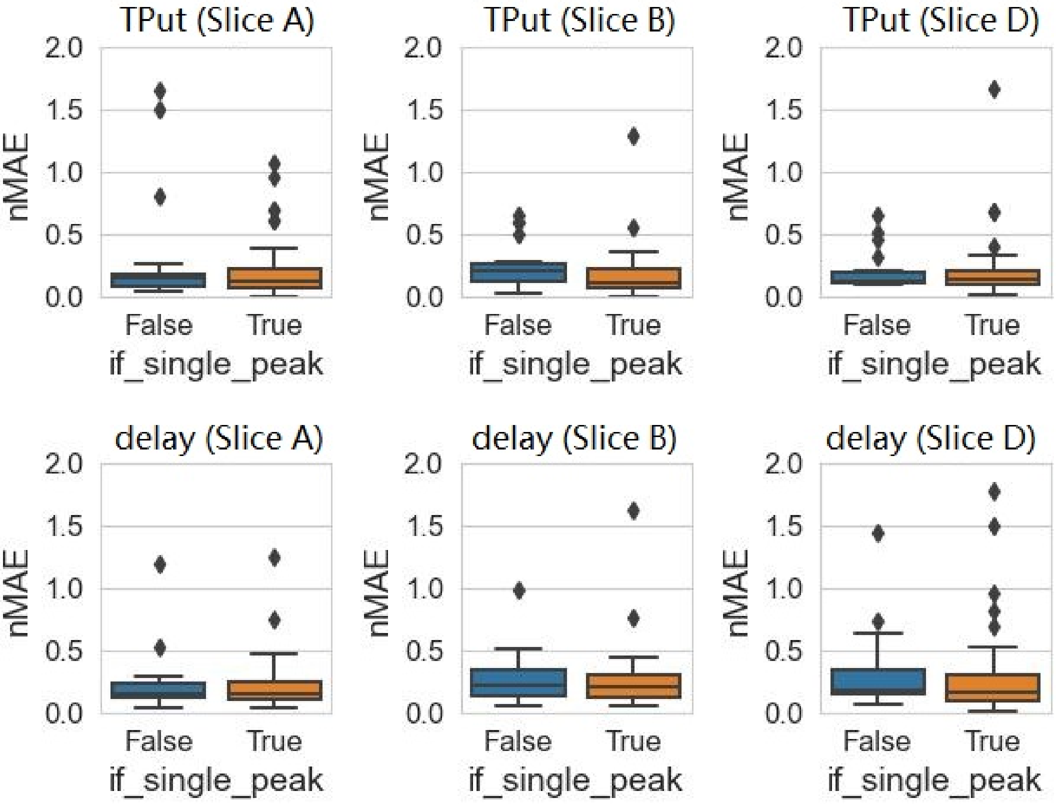


Figure 9.8: *nMAE of throughput and delay per slice.*

The nMAE values are illustrated in Fig. 9.8. It is evident that the forecasts exhibit a low nMAE. The model errors for the delay in Slice D are slightly higher than the others, but the median error remains low. These box plots suggest that a simulation that reflects reality can be achieved using VARMAX. Furthermore, it was examined whether the VARMAX model outperforms the univariate ARMA approach in simulation. For this purpose, RdR is recalculated as follows: $max(0, RdR \cdot 100\%)$. The results for all samples are shown in Fig. 9.9. Comparing the results of the VARMAX simulation with its one-dimensional ARMA counterpart reveals significant improvements in both delay and throughput for Slice A and Slice D, with negligible improvement for Slice B.

9.2.2 Simulated Scenarios

Evaluating the model allowed for the determination of its effectiveness for the specified problem. The subsequent step involves conducting simulations where the DV values are adjusted by the user. The scenarios under consideration are shown in Tab. 9.2.

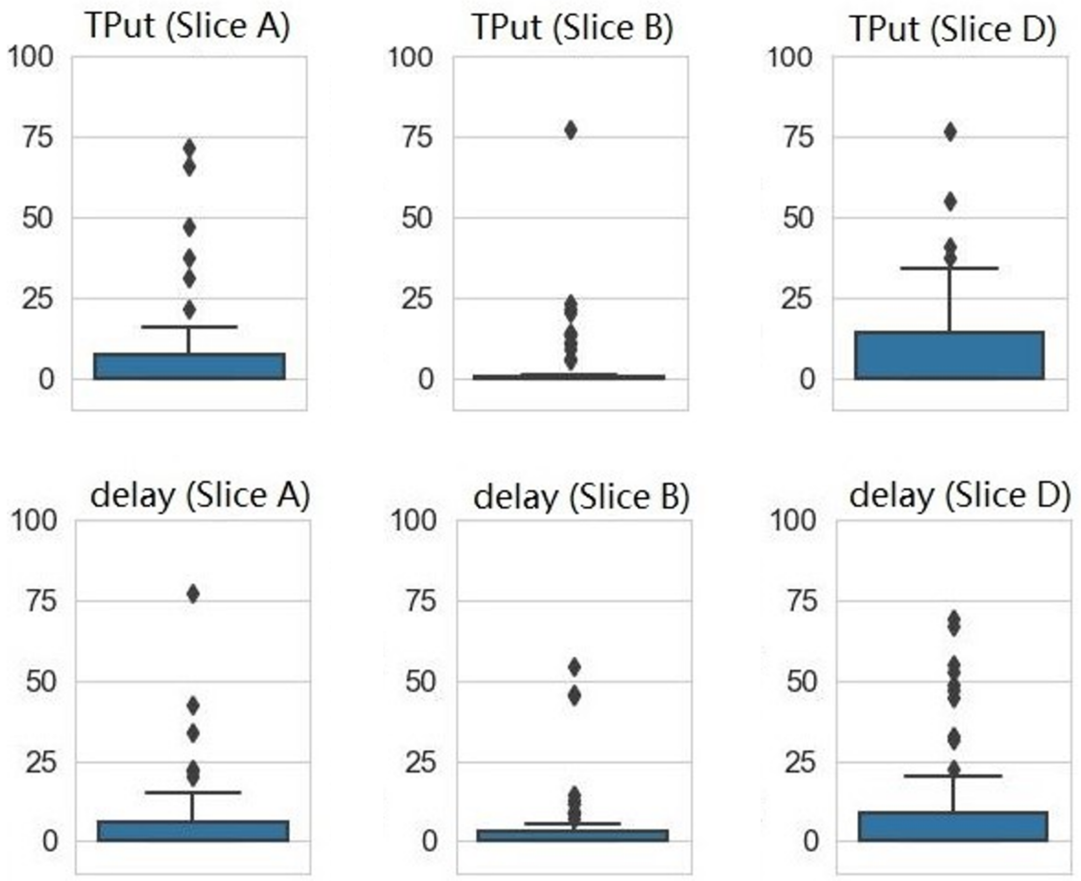


Figure 9.9: *Adjusted RdR score for all chosen samples comparing the results of the VARMAX with ARMA.*

Table 9.2: *Simulation scenarios.*

Variables	% of DV's change
DV (A), PRB utilization	50, 100, 150, 200
DV (B), PRB utilization	50, 100, 150, 200
DV (A, B, C), PRB utilization	50, 100

9.3 Summary

This chapter presents the entire 5G Network Slicing dimensioning framework. It describes also the concept of simulated scenarios and the realization of the DT. The description of each module is followed by a presentation of a single simulated scenario. Fig. 9.10 illustrates normalized throughputs and delays: comparing the last 24 hours of the training set with the forecasted values. The results pertain to the scenario where DV-A and DV-D increase by 100%. The forecast values indicate that Slices A and D have a higher

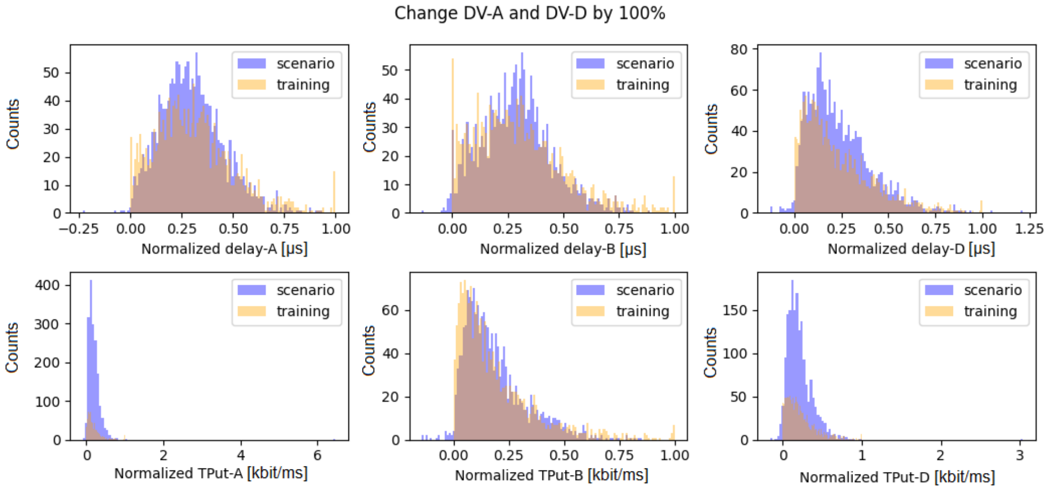


Figure 9.10: *Comparison of the histograms of normalized throughputs and delays: forecasts (simulated scenario) vs. training set (the last 24h).*

TPut and delay, in alignment with the increased DV. Slice B experiences a slight delay increase due to shared underlying resources. Additionally, the delay distribution for Slice D has widened (more samples with higher delays), attributed to its lowest priority status and the impact of all traffic. Ultimately, these findings are consistent with the understanding of this scenario in the telecommunications industry, further validating the approach.

Chapter 10

Conclusions and Future Works

10.1 Achievements and Contributions

The main achievements and contributions of this doctoral dissertation are as follows:

1. The collection and analysis of a dataset comprising hourly averaged time series data from thirty three 5G BTS operating in a live network deployment. The dataset was collected over the course of March 2023 for short-term and June 2023-February 2024 for long-term forecasting. These selected KPIs are fundamental performance indicators found in any vendor's radio equipment, facilitating the creation of multivariate models that incorporate both traffic load and radio environment metrics, which directly affect throughput and delay. The Spearman and Pearson correlation matrices revealed that many pairs of variables exhibit strongly monotonic relationships, and a strong correlation is evident for specific pairs of delays and throughputs across various network slices. The analysis concluded that seasonality and unique attributes per cell must be taken into account when choosing a forecasting model.

2. A quantitative assessment of one-dimensional models for UEs and CQI forecasting, resulting in the selection of the most accurate model. Among the models initially considered—SARIMA, Chronos, Prophet, and Lag-Llama—the last two have been selected for fine-grained comparison. The models were evaluated using MAPE and nMAE metrics, and in most scenarios, the Lag-Llama forecasting model proved to be the most precise. Furthermore, its capacity for fine-tuning positions it as a potential subject for future research.

3. A quantitative assessment of multi-variate models for slice throughput and delay short- and long-term forecasting, resulting in the selection of the most accurate one. Among the evaluated models, namely VARMAX, LSTM, CNN-BiLSTM, the

first one demonstrated best accuarcy and computational efficiency, across varying aggregation levels. Enhancements in previous short-term forecasting models have extended their applicability to forecasts of up to three months. Additionally, training the model on data from all cells within a band, as opposed to individual cells, has increased forecast accuracy. Empirical assessments using real commercial network data have validated the practicality and reproducibility of the developed forecasting models and methodologies for long-term network slicing planning.

4. Original technique, using system-level traffic data, for estimating statistical MG of aggregated 5G transport links. The proposed approach enables the scalability of the simulation outcomes, extending from a 21-cell network to 200 or more cells. The MG algorithm can use real-world measurements, focusing on actual radio performance rather than theoretical values. In addition, the proposed approach reduces the computational time from days to seconds, which is crucial for network planning recommendations and ultimately improves the efficiency and flexibility of services provided to telecommunication operators. Two case studies have been presented, demonstrating the alignment of the estimations with measured values from microwave links in mobile networks and highlighting its relevance to cloud BTS dimensioning.

5. A framework for forecasting and dimensioning, which is a data-centric model that incorporates the concept of the DT. Specifically, the framework can function autonomously as a forecasting tool applicable to (sliced) network dimensioning and traffic management, or it can operate as an integral element of a versatile DT. The framework can also be used to provide traffic forecasting based on actual network data or to support optimization systems to suggest capacity expansions or configure parameters for slice planning. Furthermore, it can be used for simulated scenarios that consider various potential changes ("what-if" scenarios) in the evolution of network traffic demand. To verify the validity of the framework and evaluate its applicability and ability to maintain the physical context, experiments were performed on the actual data.

6. Ultimately, this framework with selected models and developed algorithms has been incorporated into the company's tools to support dimensioning and planning processes.

10.2 Publications

Some ideas, achievements, considerations, figures, and tables presented in this doctoral dissertation have appeared in previously published journal articles and conference papers.

The list of all publications corresponding to the topic of the thesis is presented below in chronological order.

1. D. Dulas, K. Maraj-Zygmat, K. Walkowiak, "Method of 5G TDD Midhaul Multiplexing Gain Estimation based on System-Level Traffic Measurements", 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2022, pp. 1-6, doi: 10.23919/SoftCOM55329.2022.9911430.

2. D. Dulas, K. Walkowiak, „AI-Assisted Dimensioning of 5G Network Slices – Review and Perspectives", „Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne", 4/2023, doi: 10.15199/59.2023.4.43

3. D. Dulas, J. Witulska, A. Wyłomańska, I. Jabłoński, K. Walkowiak, "Data-Driven Model for Sliced 5G Network Dimensioning and Planning, Featured with Forecast and "what-if" Analysis", IEEE Access, vol. 12, pp. 50067-50082, 2024, doi: 10.1109/ACCESS.2024.3383324.

4. D. Dulas, J. Witulska, A. Wyłomańska, K. Walkowiak, „Data-driven model for long-term forecasting of 5G throughput and delay per network slice with the context of cell configuration", „Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne", 4/2024, doi: 10.15199/59.2024.4.93

10.3 Future Works

For future work, the following research directions are proposed:

- Assess the framework and chosen models for long-term forecasts extending beyond three months, which necessitates a dataset spanning a longer time period.

- Assess the accuracy of the one-dimensional model in forecasting environmental variables (e.g., #UEs, CQI) using Lag-Llama after fine-tuning with a larger dataset, which requires data collection from a larger network cluster.

- Investigate the generalizability of the models, potentially by incorporating configurational parameters as model features.

- Expand the initial framework version with DT for throughput and delay forecasting as a part of a project: "Development of an automatized, supervised process and system, enabling new 5G/5G+ Nokia features performance impact evaluation and recommending contextually optimal parameter settings, thanks to integration of digitalized and synthetized expert knowledge with digital twin-based simulation model", which was submitted in call for proposals: "FENG.01.01-IP.01-005/23 – Ścieżka SMART – Projekty realizowane w konsorcjach", supervised by The National Centre for Research and Development (NCBR) and is waiting for the acceptance.

Bibliography

[1] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2019.106984. URL `https://www.sciencedirect.com/science/article/pii/S1389128619304773`.

[2] M. Umar Khan, A. García-Armada, and J. J. Escudero-Garzás. Service-based network dimensioning for 5g networks assisted by real data. *IEEE Access*, 8:129193–129212, 2020. doi: 10.1109/ACCESS.2020.3009127.

[3] Xuemin Shen, Jie Gao, Wen Wu, Mushu Li, Conghao Zhou, and Weihua Zhuang. Holistic network virtualization and pervasive network intelligence for 6g. *IEEE Communications Surveys & Tutorials*, 24(1):1–30, 2022. doi: 10.1109/COMST.2021.3135829.

[4] Prakash Subramanian et al. Future x network cost economics - a network operator's tco journey through virtualization, automation, and network slicing. *Bell Labs Consulting*, 2019.

[5] Nokia Technology Strategy 2030. Global network traffic 2030 report, 2024. URL `https://www.nokia.com/technology-strategy/`.

[6] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. *IEEE Access*, 7:127639–127651, 2019. doi: 10.1109/ACCESS.2019.2939938.

[7] Wen Wu, Conghao Zhou, Mushu Li, Huaqing Wu, Haibo Zhou, Ning Zhang, Xuemin Sherman Shen, and Weihua Zhuang. Ai-native network slicing for 6g networks. *IEEE Wireless Communications*, 29(1):96–103, 2022. doi: 10.1109/MWC.001.2100338.

[8] Marcial Gutierrez et al. Ericsson's next-gen ai-driven network dimensioning solution. https://www.ericsson.com/en/blog/2022/3/next-gen-ai-driven-network-dimensioning-solution, 2022. Updated: 2022-03-23.

[9] Dominik Dulas, Katarzyna Maraj-Zygmat, and Krzysztof Walkowiak. Method of 5g tdd midhaul multiplexing gain estimation based on system-level traffic measurements. In *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6, 2022. doi: 10.23919/SoftCOM55329.2022.9911430.

[10] Anil Kirmaz, Diomidis S. Michalopoulos, Irina Balan, and Wolfgang Gerstacker. Mobile network traffic forecasting using artificial neural networks. In *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–7, 2020. doi: 10.1109/MASCOTS50786.2020.9285949.

[11] Caner Bektas, Stefan Böcker, and Christian Wietfeld. The cost of uncertainty: Impact of overprovisioning on the dimensioning of machine learning-based network slicing. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 652–657, 2022. doi: 10.1109/FNWF55208.2022.00120.

[12] Hadjer Touati, Hind Castel-Taleb, Badii Jouaber, and Sara Akbarzadeh. Split analysis and fronthaul dimensioning in 5g c-ran to guarantee ultra low latency. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4, 2020. doi: 10.1109/CCNC46108.2020.9045398.

[13] Azar Taufique, Mona Jaber, Ali Imran, Zaher Dawy, and Elias Yacoub. Planning wireless cellular networks of future: Outlook, challenges and opportunities. *IEEE Access*, 5:4821–4845, 2017. doi: 10.1109/ACCESS.2017.2680318.

[14] Jianhua Tang, Byonghyo Shim, and Tony Q. S. Quek. Service multiplexing and revenue maximization in sliced c-ran incorporated with urllc and multicast embb. *IEEE Journal on Selected Areas in Communications*, 37(4):881–895, 2019. doi: 10.1109/JSAC.2019.2898745.

[15] Qiang Ye, Weihua Zhuang, Shan Zhang, A-Long Jin, Xuemin Shen, and Xu Li. Dynamic radio resource slicing for a two-tier heterogeneous wireless network. *IEEE Transactions on Vehicular Technology*, 67(10):9896–9910, 2018. doi: 10.1109/TVT.2018.2859740.

[16] Xuemin Shen, Jie Gao, Wen Wu, Kangjia Lyu, Mushu Li, Weihua Zhuang, Xu Li, and Jaya Rao. Ai-assisted network-slicing based next-generation wireless networks. *IEEE Open Journal of Vehicular Technology*, 1:45–66, 2020. doi: 10.1109/OJVT.2020.2965100.

[17] Hoang Duy Trinh, Nicola Bui, Joerg Widmer, Lorenza Giupponi, and Paolo Dini. Analysis and modeling of mobile traffic using real traces. In *2017 IEEE 28th Annual*

International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pages 1–6, 2017. doi: 10.1109/PIMRC.2017.8292200.

[18] Jessica Moysen, Furqan Ahmed, Mario García-Lozano, and Jarno Niemelä. Big data-driven automated anomaly detection and performance forecasting in mobile networks. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–5, 2020. doi: 10.1109/GCWkshps50303.2020.9367579.

[19] Chaoyun Zhang and Paul Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks, 2017.

[20] Mathieu Leconte, Georgios S. Paschos, Panayotis Mertikopoulos, and Ulaş C. Kozat. A resource allocation framework for network slicing. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 2177–2185, 2018. doi: 10.1109/INFOCOM.2018.8486303.

[21] Gabriel Otero Pérez, José Alberto Hernández, and David Larrabeiti. Fronthaul network modeling and dimensioning meeting ultra-low latency requirements for 5g. *Journal of Optical Communications and Networking*, 10(6):573–581, 2018. doi: 10.1364/JOCN.10.000573.

[22] Amanpreet Singh, Xi Li, Indika Abeywickrama, Andreas Könsgen, Carmelita Görg, Phuong Nga Tran, and Andreas Timm-Giel. Qoe-based access network dimensioning. In *2014 16th International Telecommunications Network Strategy and Planning Symposium (Networks)*, pages 1–6, 2014. doi: 10.1109/NETWKS.2014.6959271.

[23] Oscar Adamuz-Hinojosa, Pablo Muñoz, Pablo Ameigeiras, and Juan M. Lopez-Soler. Potential-game-based 5g ran slice planning for gbr services. *IEEE Access*, 11:4763–4780, 2023. doi: 10.1109/ACCESS.2023.3236103.

[24] Raouf Abozariba, Muhammad Kamran Naeem, Md Asaduzzaman, and Mohammad Patwary. Uncertainty-aware ran slicing via machine learning predictions in next-generation networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6, 2020. doi: 10.1109/VTC2020-Fall49728.2020.9348736.

[25] Haotong Cao, Zhi Lin, Kai Sun, Chenjing Tian, Kang An, and Hongbo Zhu. Efficient slice reconfiguration for 6g networks with guaranteed qos and reduced opex. In *2024 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 473–478, 2024. doi: 10.1109/ICCC62479.2024.10681774.

[26] Aleksandra Knapińska, Piotr Lechowicz, Weronika Węgier, and Krzysztof Walkowiak. Long-term prediction of multiple types of time-varying network traffic using chunk-based ensemble learning. *Applied Soft Computing*, 130:109694, 2022.

[27] H.D. Trinh and Universitat Politècnica de Catalunya. Departament d'Enginyeria Telemàtica. *Data Analytics for Mobile Traffic in 5G Networks Using Machine Learning Techniques*. Universitat Politècnica de Catalunya, 2020. URL `https://books.google.pl/books?id=Bc0TzgEACAAJ`.

[28] Sonali Shankar, P Vigneswara Ilavarasan, Sushil Punia, and Surya Prakash Singh. Forecasting container throughput with long short-term memory networks. *Industrial management & data systems*, 120(3):425–441, 2020.

[29] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[30] Ufuk Uyan, M Tugberk Isyapar, and Mahiye Uluyagmur Ozturk. 5g long-term and large-scale mobile traffic forecasting. *arXiv preprint arXiv:2212.10869*, 2022.

[31] Maryam Mohseni, Soodeh Nikan, and Abdallah Shami. Ai-based traffic forecasting in 5g network. In *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 188–192. IEEE, 2022.

[32] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. Deepcog: Cognitive network management in sliced 5g networks with deep learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 280–288. IEEE, 2019.

[33] Saurabh Suradhaniwar, Soumyashree Kar, Surya S Durbha, and Adinarayana Jagarlapudi. Time series forecasting of univariate agrometeorological data: a comparative performance evaluation via one-step and multi-step ahead forecasting strategies. *Sensors*, 21(7):2430, 2021.

[34] Massimiliano Marcellino, James H Stock, and Mark W Watson. A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of econometrics*, 135(1-2):499–526, 2006.

[35] Arash Andalib and Farid Atry. Multi-step ahead forecasts for electricity prices using narx: a new approach, a critical analysis of one-step ahead forecasts. *Energy Conversion and Management*, 50(3):739–747, 2009.

[36] Schyler C Sun and Weisi Guo. Forecasting wireless demand with extreme values using feature embedding in gaussian processes. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–6. IEEE, 2021.

[37] Ajib Setyo Arifin and Muhammad Idham Habibie. The prediction of mobile data traffic based on the arima model and disruptive formula in industry 4.0: A case

study in jakarta, indonesia. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(2):907–918, 2020.

[38] T Tatarnikova, B Sovetov, and V Chehanovsky. Autoregressive models of network traffic prediction. In *Journal of Physics: Conference Series*, volume 1864, page 012099. IOP Publishing, 2021.

[39] Jin Wang. A process level network traffic prediction algorithm based on arima model in smart substation. In *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, pages 1–5. IEEE, 2013.

[40] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805, 2016.

[41] Jorge Martín-Pérez, Koteswararao Kondepu, Danny De Vleeschauwer, Venkatarami Reddy, Carlos Guimarães, Andrea Sgambelluri, Luca Valcarenghi, Chrysa Papagianni, and Carlos J Bernardos. Dimensioning v2n services in 5g networks through forecast-based scaling. *IEEE Access*, 10:9587–9602, 2022.

[42] Xin Dong, Wentao Fan, and Jun Gu. Predicting lte throughput using traffic time series. *ZTE communications*, 13(4):61–64, 2015.

[43] Amin Azari, Panagiotis Papapetrou, Stojan Denic, and Gunnar Peters. Cellular traffic prediction and classification: A comparative evaluation of lstm and arima. In *Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22*, pages 129–144. Springer, 2019.

[44] Chengsheng Pan, Yuyue Wang, Huaifeng Shi, Jianfeng Shi, and Ren Cai. Network traffic prediction incorporating prior knowledge for an intelligent network. *Sensors*, 22(7):2674, 2022.

[45] Leonardo Lo Schiavo, Marco Fiore, Marco Gramaglia, Albert Banchs, and Xavier Costa-Perez. Forecasting for network management with joint statistical modelling and machine learning. In *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 60–69. IEEE, 2022.

[46] Tejas Shelatkar, Stephen Tondale, Swaraj Yadav, and Sheetal Ahir. Web traffic time series forecasting using arima and lstm rnn. In *ITM Web of Conferences*, volume 32, page 03017. EDP Sciences, 2020.

[47] Xianmin Wei. Supporting vector-machine prediction of network traffic. In *2011 International Conference on Electrical and Control Engineering*, pages 3203–3206. IEEE, 2011.

[48] Jessica Moysen, Lorenza Giupponi, and Josep Mangues-Bafalluy. A mobile network planning tool based on data analytics. *Mobile Information Systems*, 2017, 2017.

[49] Aleksandra Knapińska, Katarzyna Półtorak, Dominika Poręba, Jan Miszczyk, Mateusz Daniluk, and Krzysztof Walkowiak. On feature selection in short-term prediction of backbone optical network traffic. In *2022 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2022.

[50] M Panek, I Jabłoński, and M Woźniak. Modeling configuration-performance relation in a mobile network: a data-driven approach. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2–5 September 2024, Valencia, Spain (Accepted).

[51] Jinbao Huang, Wenhao Guo, Rui Wei, Ming Yan, Yongle Hu, and Tuanfa Qin. Short-term power forecasting method for 5g photovoltaic base stations on non-sunny days based on sdn-integrated ingo-bp and rgan. *IET Renewable Power Generation*, 2024.

[52] Yair Neuman, Yochai Cohen, and Boaz Tamir. Short-term prediction through ordinal patterns. *Royal Society Open Science*, 8(1):201011, 2021.

[53] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 619 of *KDD '24*, page 6555–6565. ACM, August 2024. doi: 10.1145/3637528.3671451. URL `http://dx.doi.org/10.1145/3637528.3671451`.

[54] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K. Gupta, and Jingbo Shang. Large language models for time series: A survey, 2024. URL `https://arxiv.org/abs/2402.01801`.

[55] Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, Shirui Pan, Vincent S. Tseng, Yu Zheng, Lei Chen, and Hui Xiong. Large models for time series and spatio-temporal data: A survey and outlook, 2023. URL `https://arxiv.org/abs/2310.10196`.

[56] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. Position: What can large language

models tell us about time series analysis, 2024. URL https://arxiv.org/abs/2402.02713.

[57] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024. URL https://arxiv.org/abs/2310.10688.

[58] Yu-Neng Chuang, Songchen Li, Jiayi Yuan, Guanchu Wang, Kwei-Herng Lai, Leisheng Yu, Sirui Ding, Chia yuan Chang, Qiaoyu Tan, Daochen Zha, and Xia Hu. Understanding different design choices in training large time series models. *ArXiv*, abs/2406.14045, 2024. URL https://api.semanticscholar.org/CorpusID:270620043.

[59] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024. URL https://arxiv.org/abs/2310.08278.

[60] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 19622–19635. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3eb7ca52e8207697361b2c0fb3926511-Paper-Conference.pdf.

[61] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

[62] Hassan Halabian and Peter Ashwood-Smith. Capacity planning for 5g packet-based front-haul. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2018. doi: 10.1109/WCNC.2018.8377215.

[63] Chathurika Ranaweera, Elaine Wong, Ampalavanapillai Nirmalathas, Chamil Jayasundara, and Christina Lim. 5g c-ran with optical fronthaul: An analysis from a deployment perspective. *Journal of Lightwave Technology*, 36(11):2059–2068, 2018. doi: 10.1109/JLT.2017.2782822.

[64] Jay Kant Chaudhary, Jens Bartelt, and Gerhard Fettweis. Statistical multiplexing in fronthaul-constrained massive mimo. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–6, 2017. doi: 10.1109/EuCNC.2017.7980774.

[65] Liumeng Wang and Sheng Zhou. On the fronthaul statistical multiplexing gain. *IEEE Communications Letters*, 21(5):1099–1102, 2017. doi: 10.1109/LCOMM.2017.2653120.

[66] Shubhajeet Chatterjee, Mohammad J. Abdel-Rahman, and Allen B. MacKenzie. On optimal orchestration of virtualized cellular networks with statistical multiplexing. *IEEE Transactions on Wireless Communications*, 21(1):310–325, 2022. doi: 10.1109/TWC.2021.3095231.

[67] Mohamed Shehata, Ahmed Elbanna, Francesco Musumeci, and Massimo Tornatore. Multiplexing gain and processing savings of 5g radio-access-network functional splits. *IEEE Transactions on Green Communications and Networking*, 2(4):982–991, 2018. doi: 10.1109/TGCN.2018.2869294.

[68] Jingchu Liu, Sheng Zhou, Jie Gong, Zhisheng Niu, and Shugong Xu. Statistical multiplexing gain analysis of heterogeneous virtual base station pools in cloud radio access networks. *IEEE Transactions on Wireless Communications*, 15(8):5681–5694, 2016. doi: 10.1109/TWC.2016.2567383.

[69] Xusheng Tong, Lin Tian, Zongshuai Zhang, Qian Sun, and Yuanyuan Wang. Statistical multiplexing gain analysis for c-ran based on processing resource utilization. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 206–211, 2020. doi: 10.1109/ICCC51575.2020.9344898.

[70] Paul Almasan, Miquel Ferriol-Galmés, Jordi Paillisse, José Suárez-Varela, Diego Perino, Diego López, Antonio Agustin Pastor Perales, Paul Harvey, Laurent Ciavaglia, Leon Wong, Vishnu Ram, Shihan Xiao, Xiang Shi, Xiangle Cheng, Albert Cabellos-Aparicio, and Pere Barlet-Ros. Network digital twin: Context, enabling technologies, and opportunities. *IEEE Communications Magazine*, 60(11):22–27, 2022. doi: 10.1109/MCOM.001.2200012.

[71] Marco Polverini, Francesco G. Lavacca, Jaime Galán-Jiménez, Davide Aureli, Antonio Cianfrani, and Marco Listanti. Digital twin manager: A novel framework to handle conflicting network applications. In *2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 85–88, 2022. doi: 10.1109/NFV-SDN56302.2022.9974809.

[72] Mean Shu, Wanfei Sun, Jing Zhang, Xiaoyan Duan, and Ming Ai. Digital-twin-enabled 6g network autonomy and generative intelligence: Architecture, technologies and applications. *Digital Twin*, 2(16), 2022. URL `https://doi.org/10.12688/digitaltwin.17720.1`.

[73] Michał Panek, Adam Pomykała, Ireneusz Jabłoński, and Michał Woźniak. 5g/5g+ network management employing ai-based continuous deployment. *Applied Soft Computing*, 134:109984, 2023.

[74] Swati Roy, David Applegate, Zihui Ge, Ajay Mahimkar, Shomik Pathak, and Sarat Puthenpura. Quantifying the service performance impact of self-organizing network actions. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 37–45, 2016. doi: 10.1109/CNSM.2016.7818398.

[75] Yuan Su, Haoyuan Cheng, Zhe Wang, Junwei Yan, Ziyu Miao, and Aruhan Gong. Analysis and prediction of carbon emission in the large green commercial building: A case study in dalian, china. *Journal of Building Engineering*, 68:106147, 2023. doi: https://doi.org/10.1016/j.jobe.2023.106147.

[76] Jacek Koronacki and Jan Mielniczuk. *Statystyka: dla studentów kierunków technicznych i przyrodniczych*. Wydawnictwa Naukowo-Techniczne, 2001.

[77] Dominik Dulas, Justyna Witulska, Agnieszka Wyłomańska, Ireneusz Jabłoński, and Krzysztof Walkowiak. Data-driven model for sliced 5g network dimensioning and planning, featured with forecast and" what-if" analysis. *IEEE Access*, 12: 50067 – 50082, 2024.

[78] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[79] Alexander Zlotnik, Juan Manuel Montero-Martínez, and Ascensión Gallardo-Antolín. A comparison of multivariate sarima and svm models for emergency department admission prediction. In *International Conference on Health Informatics*, 2013. URL `https://api.semanticscholar.org/CorpusID:28080256`.

[80] Stylianos I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis. Comparison of sarimax, sarima, modified sarima and ann-based models for short-term pv generation forecasting. In *2016 IEEE International Energy Conference (ENERGYCON)*, pages 1–6, 2016. doi: 10.1109/ENERGYCON.2016.7514029.

[81] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell.

Stan : A probabilistic programming language. *Journal of Statistical Software*, 76, 01 2017. doi: 10.18637/jss.v076.i01.

[82] Peter J Brockwell and Davis. *Time series: theory and methods.* Springer Series in Statistics, 1991.

[83] Ian T Jolliffe. *Principal component analysis for special types of data.* Springer, 2002.

[84] James Douglas Hamilton. *Time series analysis.* Princeton university press, 1994.

[85] Matt Chapman. *A Meta-Analysis of Metrics for Change Point Detection Algorithms.* Spring, 2017.

[86] Slawek Smyl. Cognitive toolkit helps win 2016 cif international time series competition, Oct 2016. URL `https://learn.microsoft.com/pl-pl/archive/blogs/machinelearning/cognitive-toolkit-helps-win-2016-cif-international-time-series-competition`.

[87] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[88] Yu Hen Hu and Jenq-Neng Hwang. *Handbook of neural network signal processing.* CRC press, 2018.

[89] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[90] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[91] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

[92] Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, 3 (3):196–207, 2020.

[93] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A comparison of lstm and gru networks for learning symbolic sequences. *arXiv preprint arXiv:2107.02248*, 2021.

[94] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[95] Justyna Witulska. Multidimensional predictive modeling for 5g network dimensioning. Master's thesis, Wrocław University of Technology, 2023.

[96] Ruoyu Sun. Optimization for deep learning: theory and algorithms. *arXiv preprint arXiv:1912.08957*, 2019.

[97] Chitra Desai. Comparative analysis of optimizers in deep neural networks. *International Journal of Innovative Science and Research Technology*, 5(10):959–962, 2020.

[98] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

[99] Devin Knight, Erin Ostrowsky, Mitchell Pearson, and Bradley Schacht. 2022.

[100] Amanpreet Singh, Indika Abeywickrama, Andreas Könsgen, Xi Li, and Carmelita Goerg. Statistical analysis of traffic aggregation in lte access networks. In *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–4, 2013. doi: 10.1109/WMNC.2013.6548996.

[101] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. doi: 10.1214/aos/1176344552. URL `https://doi.org/10.1214/aos/1176344552`.

[102] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans.* Society for Industrial and Applied Mathematics, 1982. doi: 10.1137/1.9781611970319. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611970319`.

[103] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1994. ISBN 9780412042317. URL `https://books.google.pl/books?id=gLlpIUxRntoC`.

[104] S. N. Lahiri. *Resampling Methods for Dependent Data.* Springer Series in Statistics, 2003.

[105] Dimitris N. Politis and Joseph P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994. doi: 10.1080/01621459.1994.10476870.

[106] Anthony Christopher Davison and D. V. Hinkley. *Boostrap methods and their applications.* Cambridge University Press, New York, 1997. ISBN 0521574714 9780521574716 0521573912 9780521573917. URL `http://www.amazon.com/Bootstrap-Application-Statistical-Probabilistic-Mathematics/dp/0521574714`.

[107] Halbert White and Dimitris Politis. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23:53–70, 12 2004. doi: 10.1081/ETC-120028836.

[108] Andrew Patton, Halbert White, and Dimitris Politis. Correction to “automatic block-length selection for the dependent bootstrap” by d. politis and h. white. *Econometric Reviews*, 28:372–375, 01 2009. doi: 10.1080/07474930802459016.

[109] K. Sheppard. bashtage/arch: Release 4.18 (version v4.18). https://doi.org/10.5281/zenodo.593254, 2021.

[110] Harvard Business R. Enhancing innovation in telecom with digital twins. https://hbr.org/sponsored/2022/03/enhancing-innovation-in-telecom-with-digital-twins, 2022. Updated: 2022-03-23.

[111] Yiwen Wu, Ke Zhang, and Yan Zhang. Digital twin networks: A survey. *IEEE Internet of Things Journal*, 8(18):13789–13804, 2021. doi: 10.1109/JIOT.2021.3079510.

[112] Dave Cote. Rdr score, 2020. URL `blog/RdRscoreatmasterÂůCoteDave/blogÂůGitHub`.