Capacity planning by Network Traffic Prediction

February 16th 2024

MSc Thesis

Alexandra Verhagen

Delft University of Technology

Capacity planning by Network Traffic Prediction

February 16th 2024

by

Alexandra Verhagen

to obtain the degree of Master of Science
at the Delft University of Technology

Thesis committee:	Dr.ir. E. Smeitink	TU Delft, supervisor
	Dr. J. Söhl	TU Delft
	MSc D. Tuinhof	KPN, supervisor
	Dr. E. van Boven	TU Delft
Project Duration:	February, 2023 - February, 2024	
Faculty:	Faculty of Electrical Engineering, Delft	
Student number:	4475593	

Preface

This thesis marks the end of my master's degree in Wireless Communication & Sensing at the faculty of Electrical Engineering at the Delft University of Technology.

Foremost, I would like to express my gratitude to my two supervisors, Eric Smeitink and Denice Tuinhof. Their support and weekly guidance have been invaluable throughout the entire process. Their expertise and encouragement have significantly contributed to the development and completion of this work. I am grateful for their dedication and time, which allowed me to create a thesis that I can take pride in.

I would also like to extend my gratitude to Edgar van Boven, whose assistance in finding the project and continuous interest in its progression ensured positive strides. Moreover, my appreciation is extended to Edo Pappot, who alongside Denice, played a crucial role in the creation of this research. His enthusiasm and experience in the industry have been a source of motivation throughout this journey. Additionally, I also wish to thank Marco de Wilde and his team for their insightful views and collaboration. Lastly, I want to thank Jakob Söhl for his valuable contribution as part of my committee.

I am deeply grateful to my family, friends and boyfriend for their support, understanding and encouragement. Their company provided not only support during challenging times in the past year but also brought moments of fun and relief, including much needed holidays.

As this chapter of my academic journey concludes, I am grateful for the exciting journey Delft has brought me and the memories made in the past years.

Alexandra Verhagen
Rotterdam, February 2024

Abstract

Accurate capacity planning is essential to ensure uninterrupted services and network stability through peak hours for the transport core network of KPN. This involves a trade-off between minimizing the risks of capacity shortages and costs of capacity expansions. High network loads are occurring more frequently and their magnitude is increasing. This necessitates measures to foresee high load situations before network capacity is surpassed. Currently, planning is based on manual predictions that lack substantiation. This research aims to improve network capacity planning by development of a forecast for the next year.

An analysis of the daily maximum traffic data of the transport core is performed, to determine the most suitable models for the prediction of network traffic. The data analysis, employing time series decomposition, revealed non-stationary trends and annual seasonality; traffic decreases throughout the summer and increases in the winter. An upward trend in the frequency and intensity of traffic peaks, highlights the growing demand and shifts in usage behavior. The extreme traffic peaks in the historical data were correlated to F1 race days and other anticipated events.

Two algorithms that integrate exogenous variables were assessed to predict the extreme values. The models either yielded inaccurate traffic predictions or encountered challenges in interpretability and pattern recognition, with the limited amount of data available. In response to these limitations, a decomposed forecast was created that predicts the trend and seasonality. Furthermore, Extreme Value Analysis (EVA) was implemented to address the extreme values in the data.

The final prediction framework combines the decomposed forecast with EVA for the next six quarters and outperforms the other models. The model effectively captures extreme values and provides insights into the maximum expected peaks and risk levels. The substantiated forecasts of the EVA model and the manual predictions yielded comparable results. However, the EVA model provides better insights into the likelihood of exceeding specific traffic values, which enhances capacity calculations and precision.

The prediction framework has been integrated into the business interface of KPN, which marks the initial step in the automatization of short-term capacity planning. The research insights emphasize the intricate nature of accurate prediction of future demand and advocate for scalable solutions beyond building new capacity. These solutions range from short-term mitigation to long-term strategies designed to alleviate high network loads. They underscore the importance of the implementation and integration of dynamic decision-making within a digital twin of the network to ensure sustained effectiveness.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
ACF	Autocorrelation function
ADF	Augmented Dicky-Fuller
AI	Artificial Intelligence
AR	Auto-Regressive
AR/VR	Augmented Reality / Virtual Reality
ARIMA	Auto-Regressive Integrated Moving Average
CapEx	Capital Expenditure
CDN	Content Delivery Network
DSLAM	Digital Subscriber Line Access Multiplexer
DNS	Domain Name System
EVA	Extreme Value Analysis
FTTH	Fiber To The Home
GARCH	Generalized Auto Regressive Conditionally Heteroscedastic
GPD	Generalized Pareto Distribution
GRU	Gated Recurrent Unit
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
IQR	Interquartile range
i.i.d.	Independent and identically distributed
KPSS	Kwiatkwoski-Phillips-Schmidt-Shin
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
MAPE	Mean Absolute Percentage Error
Mbps	Megabits per second
MLE	Maximum Likelihood
MSE	Mean Squared Error
OLT	Optical Line Terminal
PACF	Partial Autocorrelation function
POP	Point of Presence
RTSP	Real Time Streaming Protocol
QoS	Quality of Service
RNN	Recurrent Neural Network
SARIMA	Seasonal ARIMA
STB	Set-top Box
Tbps	Terabits per second
UMTS	Universal Mobile Telecommunication System
VOD	Video on demand
4G	Fourth-generation
5G	Fifth-generation

1 Introduction

1.1. Data traffic growth

The use of the Internet has become an indispensable part of life. Through the convenience of mobile devices, the ubiquity of wireless networks, and the reliability of high-speed fixed broadband connections, people can access online services from anywhere. Due to advancements in technology, such as fourth-generation (4G) or fifth-generation (5G) connectivity and fiber optics, faster network speeds have become attainable. With the increased popularity of smart devices, new trends, and higher network speeds, the demand for network capacity grows rapidly. As more and more people rely on telecommunication, it is expected that the demand for data traffic will continue to grow at an unprecedented rate [1]. To keep up with this demand, the telecommunications sector has been continuously innovative.

At present, the majority of network traffic can be attributed to video services. Multiple applications have emerged in recent years for video on demand (VOD) content and real-time event streams. Due to the widespread accessibility of 4G and 5G, people have been provided with increased flexibility to watch sports or television programs on their mobile devices from any location. Additionally, operators provide interactive television, which creates other data traffic on the network (more on this is described in Section 2.2). Moreover, in the aftermath of the global COVID-19 pandemic which significantly changed the dynamics of everyday life, hybrid work has become the current norm. The necessity of remote work due to health concerns and lockdowns prompted rapid adjustments by telecom companies to support work from home. This shift not only reshaped the professional landscape but also induced a significant impact on the usage behavior of internet services. The need for video conference applications such as Zoom, resulted in a ten times increase in usage [2].

Currently, companies are in the midst of a digital transformation. New technologies such as cloud computing, Artificial Intelligence (AI) and Internet of Things (IoT) provide companies with more advanced and adaptable infrastructure options [3]. This digital transformation is evident in various markets and with that, new challenges arise for network operators. High bandwidth services such as cloud gaming or Augmented Reality, Virtual Reality (AR/VR) will demand a robust and efficient network infrastructure that meets the growing consumer demand. Until now, increased traffic loads on the network have mainly been caused by (high-quality) video streams. With these new technologies on the horizon, it is expected that other high-bandwidth services may become the main cause of increased demand. In the future, AR/VR or cloud services could significantly impact the usage behavior of internet services and the required capacity of the network. Therefore, efficient digitalization strategies are essential for the continued growth and evolution of wireless communication systems. These strategies aim to optimize network operations and deliver high-speed, reliable connectivity to users.

Network capacity planning

High bandwidth services lead to an increased load on the network and emphasize the need for a robust telecom infrastructure. In addition, the dynamic usage behavior in network traffic underscores the critical importance of this. In this dynamic landscape, telecom operators have to adapt to technological advancements and usage trends. Hence, it is necessary to implement strategies for innovation and optimize processes. To be able to handle services that demand high bandwidth, capacity management is performed to provide high-quality experiences to users.

Capacity management revolves around a systematic measurement of the volume of traffic on equipment and strategic expansion of capacity when required. Anticipation of high traffic load and risks in the network, are significantly involved in the decision-making of capacity planning [4]. It holds an essential role in the enhancement of operational efficiency within capacity management. Through close monitoring and analysis of the traffic load on the network, capacity planning ensures that the network infrastructure can scale to meet future requirements. This approach helps mitigate the risk of outages in the services provided and sustain a high Quality of Service (QoS) to end users [5]. To realize this, insights into current and future demand are required. Estimates on future demand are needed, as decisions on changes in the network infrastructure have to be made well in advance. Consequently, a prediction of the network throughput is necessary. To achieve this, historical data on traffic usage must be understood to generate a prediction. This analysis combined with an accurate forecast, is highly valuable to make timely investments in the network and deliver uninterrupted daily services to clients.

1.2. The KPN network

One of the biggest telecom providers in the Netherlands is KPN. The company offers a wide range of services for telephony, data, and television on its fixed and mobile networks. KPN has over 10 million mobile subscribers and offers services to over 4 million broadband customers [6]. The fixed and mobile network of KPN is considered to be one of the biggest networks compared to other Dutch providers.

The transport core network of the company is comprised of fiber optics. At present, the company is expanding the fiberization from the central hubs in the transport core to the last-mile access. The last mile access refers to the last components in a telecom network to reach the end user. This includes fiber optics that extend to the distribution cabinet, but also fiber optics that extend to houses of customers, or Fiber to the Home (FTTH). The fiber roll-out aims to have 80% of the Netherlands covered in fiber optics in 2026. This network of fiber increases the possible bandwidth provided to users of the network.

The Netherlands is divided into 161 areas and each area is served by a Metro Core (MC) location. These are network interface points, also known as Points of Presence (POP), where the fiber optic cables in a particular area converge. The data is then transmitted through distribution points to other POPs or end users. These distribution points are referred to as Digital Subscriber Line Access Multiplexers (DSLAM), for copper, or Optical Line Terminals (OLT), for fiber. These DSLAMs and OLTs are connected to the local distribution cabinets that connect the network to the end users. All of these stations are connected and form the network of KPN. The high-level structure of the network is illustrated in Figure 1.1 and comprises three layers; the service layer, the transport core, and the access layer.

The service layer consists of various domains in the network that offer services for voice and data communications. More on the service layer and the domains that are provided is described in Section 2.1. All of the network cables used to deliver services from the service platforms to the end user are connected via the transport core.
The transport core serves as the primary infrastructure of the network and is composed of high-capacity transmission lines and switches. It is the most vital part of the network and manages the highest capacity load. The traffic to the transport core originates from the service platforms that are connected to this transport core. It aggregates all the data of the service platforms and distributes the data to the access nodes. Moreover, all the internet traffic from the connection between the global internet exchange and the network passes through this core.
The access layer is the part of the network that connects customers to the services of the transport core. For mobile services, the access network includes cellular towers that provide coverage. Access is distributed across the country with 161 MC locations. Every MC connects a network area of the Netherlands to Metro Bridge (MB) nodes. These are connected through access nodes to all DSLAMs and OLTs that provide the connections to users.

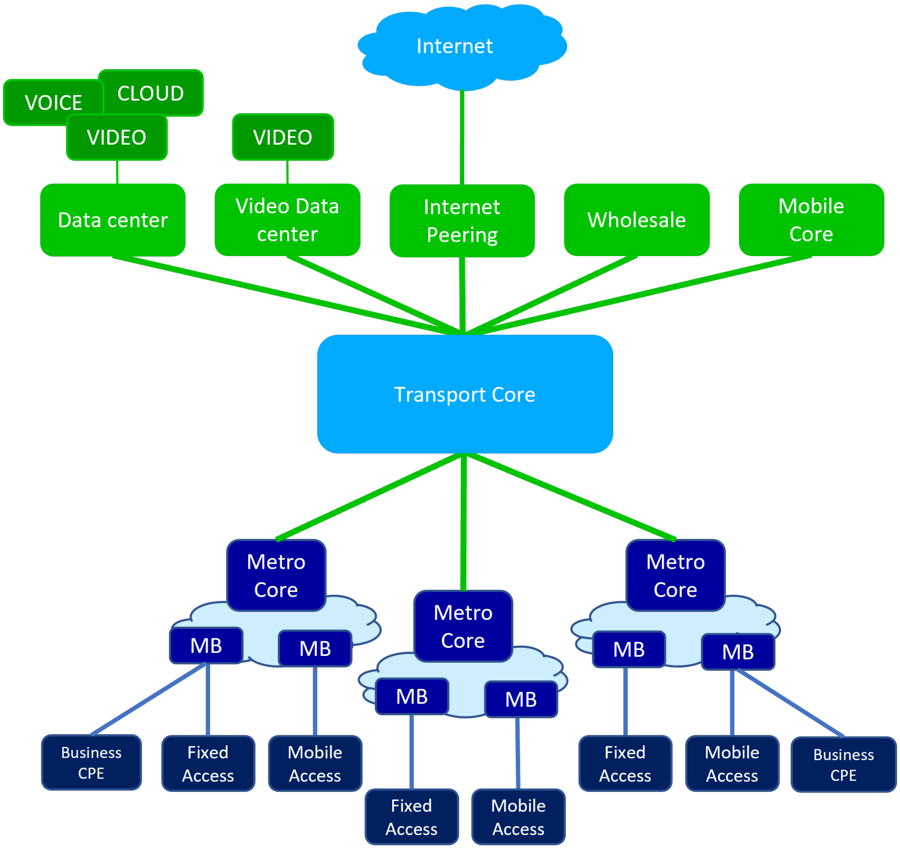


Figure 1.1: High-level view of the network

The transport core has used fiber optics for transmission in the past decades. Before the use of fiber optics in the access layer, copper cables were the primary mode to transmit data. The capacity of copper cables has improved to transmit larger amounts of data. Despite evolution to accommodate newer technologies, the distance to the transport core remains a challenge for copper cables. Fiber optics is a more recent technology in which data transmission is provided by cables that consist of thin strands of glass. Data is transformed into a pattern of light signals with different wavelengths, which enables simultaneous data transmission on multiple channels. Fiber optics are less susceptible to interference and maintain a strong signal over longer distances. This results in higher capacity transmission possibilities compared to copper cables. Nowadays, telecom operators use fiber optics more and more to transmit data to the last mile access.

The roll-out of fiber optics in the access layer provides higher speed per user. However, if all users employ high bandwidth services on the network simultaneously, bottlenecks in the transport core can occur. To keep up with the enormous increase in end-user speeds due to the mass deployment of fiber, the capacity of the transport core will also need to increase. Therefore, the emphasis of this research is on capacity planning for the transport core, as interruptions in the transport core have the highest impact on the network. A shortage of capacity here can lead to service interruptions for the whole country. Hence, the goal is to proactively assess the required capacity to minimize the risk of capacity shortages.

1.3. Problem definition

To provide continuous services to customers and maintain a stable network in peak hours, a safe margin for network traffic capacity is designed into the network. If this safe margin is exceeded, a network outage will occur and no services will be available for customers. It is vital to prevent outages at all times, as they diminish network reliability, have financial repercussions and will result in reduced customer satisfaction. Additionally, to build extra capacity in the transport core involves long lead times and high costs. Investments in overcapacity and excessive infrastructure must be avoided. Capacity planning is a trade-off between minimizing the risk of capacity shortages and strategic investments. To tackle this challenge, it is important to consider that high network loads will occur more often and the magnitude of this load is increasing. This is apparent in the historical data of the network throughput. Moreover, due to the rapid roll-out of fiber, the total amount of traffic and the magnitude of extreme loads in traffic will increase. It is essential to foresee this high load before traffic will exceed the capacity of the network. To address the question of the demand necessary for capacity planning, predictions of network traffic are computed.

Currently, network traffic predictions for the next four quarters are performed manually, which involves numerous parameters for complex equations. The calculations of these forecasts occasionally lack adequate substantiation, whilst clarity is essential for business purposes that concern significant investments. This forecast can be validated and challenged by a model, that considers other sources for predictions, particularly an analysis of the historical data and its outliers. Furthermore, this model is the initial phase of an automated capacity planning process. Hence, this project will focus on the improvement of capacity planning, provide predictions and model events with an unexpectedly high load, for which a manual estimation does not suffice. This is to ensure continuity and reliability in the services provided to customers.

1.4. Research questions

The main objective of this research is to improve the forecast of network traffic and implement an automated model for capacity planning purposes. Based on the research insights, potential recommendations are considered for implementation in the network infrastructure. The main research question is:

How to forecast network traffic including extreme values in order to improve and automate network capacity planning?

To help answer the main research questions, the following sub-research questions arise:

1. What is the underlying cause of the extreme network traffic peaks observed in the historical data?
2. Is it possible to use a machine learning algorithm to create a network traffic prediction model with exogenous variables, that predicts extreme values?
3. Is it possible to use Extreme Value Analysis to forecast network traffic peaks, including the risk of exceeding a certain level, and how can this be implemented?
4. How does the prediction model perform compared to the manual prediction method that has been used until now?
5. How could the prediction model and the insights in network traffic from this research contribute to new solutions in network design?

1.5. Thesis synopsis

The process of this study is described in the next chapters. Chapter 2 describes more background on the transport core, the current situation of capacity planning and the design requirements for implementation of the model. In Chapter 3, previously done research on this matter is described and the approaches considered relevant for this project. Chapter 4 explains the analyzed data and its characteristics. This data is used as input for the possible prediction models, for which the methodology is described in Chapter 5. In this chapter, a machine learning model and statistical approaches are compared. Thereafter, the final framework for the prediction model and its results are explained in Chapter 6. Additionally, Chapter 7 elaborates on the implementation of the model in the business interface for real-time usage. Moreover, possible recommendations for the network and mitigation solutions are described in Chapter 8. Lastly, Chapter 9 concludes the results of this study, a discussion of the research and the opportunities for future development.

2 Capacity planning for KPN

This chapter describes background knowledge on the transport core, current capacity planning, and design requirements. Moreover, the impact of this research is explained. Lastly, more elaboration is given on the implementation of the design of the framework to ensure that the model is usable for business purposes.

2.1. Transport core

The core layer of the network, the transport core, comprises four data centers. The transport core data centers are located in the cities of Zwolle, Arnhem, Rotterdam, and Amsterdam, the so-called ZARA locations. The optic fiber lines that provide the transmission between these four data centers consist of multiple 100G transmission lines. The four data centers of the transport core are each connected to all other ZARA locations. This is to ensure continuous services for each data center, in case one of the connections or ZARA locations fails.
The transport core is the backbone of the network, which facilitates the transport of the network traffic between the service layer and the access layer. It acts as a central hub where the traffic from the service domains is aggregated and transported across the entire network. The time series covers the period from March 9, 2020, to October 1, 2023. It includes 1302 days of network traffic data; traffic is measured in five minute intervals and the maximum peak per day is stored. The time series of Mobile Core is of shorter length due to the delayed availability of measurements of this service domain.

The traffic of the Data Center and Video Data Center were previously part of a single Data Center domain. The Video Data Center was created due to the implementation of a new protocol for the Content Delivery Network (CDN) of the Data Center. More on this is elaborated on in the functionalities of the domains in the service layer below:

- **Data Center**: Various service platforms are connected to the Data Center, for television, voice, and cloud applications to be accessible to customers.
  Firstly, the core of the voice network is provided here, which enables users to make phone calls. Also provided here is the Domain Name System (DNS), which assigns domain names to IP addresses to help users access websites and other services.
  The CDN provides real-time live streams of television programs via multicast and VOD (non-live) streams via unicast. As will be elaborated on in Section 2.2.1, popular VOD content is available from decentralized CDNs at the access layer. Less popular VOD content is provided at the service layer. The CDN follows two protocols; the first protocol is provided from this domain and the second protocol is provided from the Video Data Center. In the future, all streams will follow HTTPS.
- **Video Data Center**: The requested television streams of the second protocol and streams of content applications provided on the network of KPN, are accessed through this domain. For example, customers who request less popular video content and other Internet content via fixed Internet, will generate traffic through this domain. Popular video content is again decentralized to the access layer.
- **Internet Peering**: This domain contains all further traffic from the Internet and other content applications, accessed on the fixed network. Furthermore, it contains the traffic load from the mobile networks. This is because the mobile networks are built on the fixed network and therefore,

the throughput traverses from the Mobile Core domain to the Internet peering domain. Hence, a part of the throughput from this domain reflects the data generated by unicast streams from mobile devices.

- **Wholesale**: All Wholesale Ethernet Access Services for Wholesale customers, thus other companies that offer services by using the KPN Network, are accessed from this domain.
- **Mobile Core**: Mobile data and voice communications are connected to the mobile core. All streams of mobile data are visible in the throughput of this domain.

2.2. Current situation

Capacity planning is crucial to ensure that the network evolves with the larger traffic demand. Failure to anticipate high traffic loads on the network can result in expensive scale-up efforts or, in the worst case, network outages. Implementation of capacity expansion plans in the fixed transport core requires time. Ergo, investments in the network infrastructure must be made at least six months in advance. Consequently, predictions are computed to make well-informed decisions on this matter in time. It is a continuous process for fixed core capacity management and involves many phases, as illustrated in Figure 2.1.

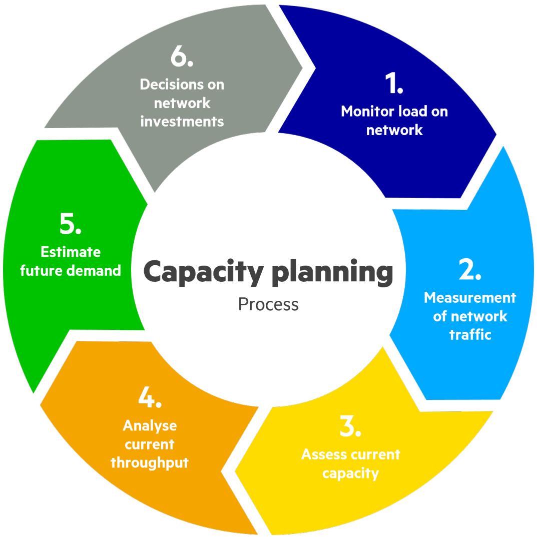


Figure 2.1: Process of capacity planning for the transport core.

1. Initially, the load on the network is collected through monitoring tools.
2. These measurements are updated every five minutes and allow for real-time monitoring of network load. This data is temporarily stored for continuous observation of network traffic over the past 24 hours. The throughput is measured every five minutes and the maximum throughput value recorded within that day, is saved for additional insights.
3. Followed by this, the current capacity of the physical equipment is assessed.
4. The current throughput is then analyzed and the capacity is compared with the measured throughput.
5. Subsequently, the daily peak values of the historical and current throughput are used to estimate future demand.
6. Based on these numbers, decisions are made on where and when to expand capacity in the fixed transport core network. This is a trade-off between cost and efficiency for capacity expansions. The unpredictability of network traffic adds complexity to this process. Moreover, the duration of construction and installation varies based on the complexity of the expansion. For instance, the construction of a server cabinet demands more time compared to the installation of free slots on a current network module.

With this knowledge, strategic network investments are made. This is an iterative process; it is evaluated and repeated to meet the larger traffic demands to keep the network stable. The current strategy requires that the network must be built with a sufficient amount of extra capacity, to handle high network traffic loads in potential worst-case scenarios. These scenarios are designed for the worst-case situations, in which network errors across various components result in the most significant disruptions. The network must be able to handle the maximum amount of load that has been recorded on the links, with the effects of a worst-case scenario. These redundancy measures are designed into the network to ensure that the network remains stable. However, data traffic grows and to build excessive capacity expansions on potential worst-case scenarios is very expensive. This could be improved more cost-effectively when further insights are available into the future trends and usage behavior of customers.

Currently, estimates on future demand are computed manually. Various factors on technical developments, market share and bit rate, are taken into account to determine the required capacity in the future. These factors include fiber-optic rollout, statistics from external parties and expected releases of video or gaming applications. The capacity planning team of the company uses two types of forecasts at the moment, which are a long-term and short-term estimation of the throughput. The long-term forecast estimates the growth of the traffic in three years or beyond and is required for strategic plans over a longer period. The short-term forecast is the prediction of the required future demand in the next four quarters.

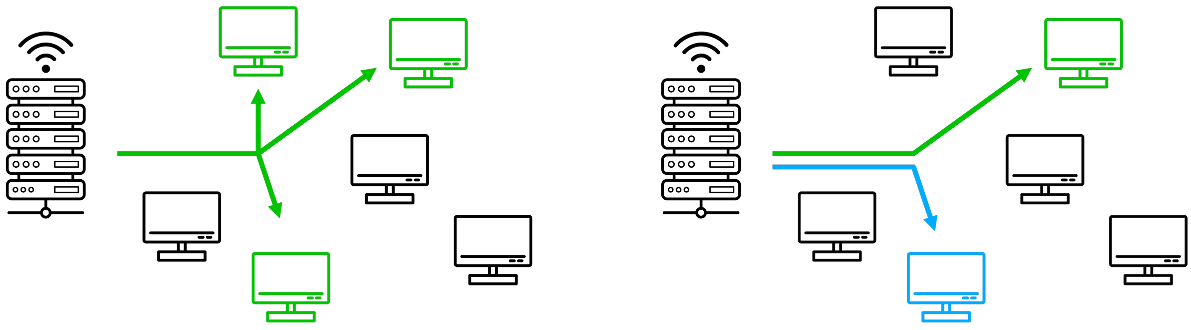
This research aims to improve and automate the short-term forecast process. The short-term forecast should be calculated for six quarters in advance and should give the maximum expected throughput per quarter. Specifically, it has to describe the expected peak value that will be reached in one day on a specific domain in the network. Therefore, the daily peak value is considered the first design requirement for the model.

At present, the maximum expected throughput is calculated for all different domains within the service layer, which are linked to the transport core. However, the short-term forecast is computed every quarter for the next four quarters instead of six quarters. This is because annual budgets are set for the network infrastructure, which determines the investments possible for scaling up capacity every year. Hence, the second design requirement is to produce a one year forecast. Nevertheless if feasible with the dataset, the extension of the forecast to six quarters would be preferable.

2.2.1. Decentralization of content

Followed by the insights gained from capacity management, some measures have been taken to reduce the traffic load in the transport core. This traffic was generated by applications in the service layer, which caused high loads on the network. As a result, some content has been decentralized and is now delivered by a CDN at the access layer. This approach eliminates the need for traffic to traverse from the service layer through the transport core before it reaches the user. Hence, when users request the content, the load is eliminated from the transport core and content can be accessed more efficiently. This content comprises popular VOD content; non-live television programs, sports events that have occurred and video content, all of which can be cached from a different location in the network. Due to the high latency of current technologies that would occur when caching live content, decentralization of content is only possible for programs that are not streamed live.

Live coverage of (sports) events employs different protocols for casting, which have been illustrated in Figure 2.2. Firstly, live events can be viewed on television via a set-top box (STB) that employs multicast or one-to-many communication. Multicast is an efficient transmission method, where one dedicated stream is distributed to multiple users to deliver video content. When multiple users request the same content, the content server sends the packets once with a bitstream of, for instance, 5 Megabits per second (Mbps). The switches and routers forward content only to the hosts who have requested it. Therefore, only one stream of 5 Mbps is required for this process, which is depicted in Figure 2.2a. When live television is paused for a short time and started again, the connection to the dedicated multicast stream is interrupted and the transmission mode shifts to unicast. Unicast, or one-to-one transmission, is an alternative method that delivers content through unique connections as shown in Figure 2.2b. Unicast is the primary method for streaming VOD content. When a user streams content from their mobile device, a unicast connection is established in the network. If numerous users simultaneously stream from their mobile devices, this results in multiple streams that require 5 Mbps per user. This means that each user creates a unique connection rather than the usage of the live television multicast stream, which places a substantial load on the network.



(a) Multicast transmission

(b) Unicast transmission

Figure 2.2: Two types of transmission employed to stream content to hosts.

2.3. Impact

As stated before, the load on the network is monitored by measurement of network data. This data collection is used to analyze the throughput of the data throughout the day. From the measurements, it is evident that the general trend of data is larger and network loads are bigger. Since the breakout of COVID-19, there has been a 40% increase in data traffic in 2021 compared to February 2020 [7]. The measurements also indicate that popular large-scale events are visible in the network data, such as the Eurovision Song Contest or UEFA Euro matches. On both fixed and mobile networks peaks are visible in the throughput, as more people watch events on their mobile devices. Insights from discoveries like these are crucial for capacity planning and show the significance of the information that can be gained from measured data. An analysis of historical data usage provides additional valuable perspectives and is further explored in Chapter 4.

Formula 1 (F1) has gained significant popularity in the Netherlands and its races have become among the most watched and streamed sports programs. Up to 60% more mobile download traffic than usual was recorded in 2021 in the Grand Prix that secured the first Dutch F1 World Championship title. Again, the data traffic numbers of the network indicate the occurrence of F1 races, which is evident from the peaks in those moments. Figure 2.3 shows the mobile data traffic during two Sundays. An increase of 50% is visible on the Sunday with an F1 race compared to a usual Sunday.

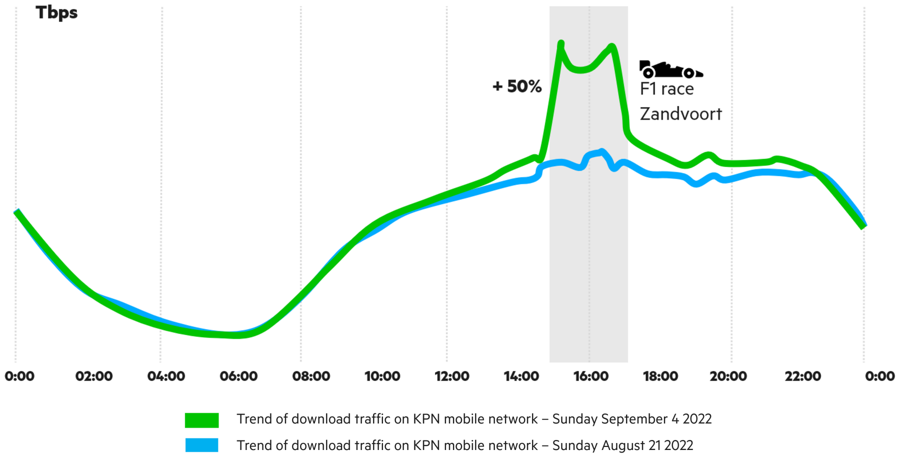


Figure 2.3: Comparison of mobile data traffic in Terabits per second (Tbps) during two Sundays in 2022 [8].

In March 2022, the streaming service Viaplay and KPN launched their distribution deal. This partnership made the streaming service accessible to 3.6 million households in the Netherlands. Within this group of potential viewers, Viaplay provides multicast transmission on the fixed network of KPN to customers to watch live content on their television channels. This network design decision minimizes the load on the transport core for this type of casting. Without this approach, the multicast streams would have been thousands of additional unicast streams in Mbps. This would have placed an excessive load on the transport core, for which there would be insufficient capacity.

2.4. Contribution

Firstly, this research marks the initial phase in the design of a dynamic forecast model that uses historical data as input. The intention is to integrate the model as a microservice for a digital twin of the network and will be focused on capacity forecasts for network data traffic. A digital twin is a virtual representation of a system or a process, that mirrors the behavior and characteristics of the physical environment [9]. This concept integrates data from various sources, which includes IoT devices, network equipment and operational processes. As a result, a dynamic and comprehensive model is created of the behavior of the actual network. This model can provide valuable insights into network performance under certain situations, as it enhances the ability to monitor, analyze and respond to possible real-time dynamics within the network. It allows operators to anticipate and address risks or issues before they can impact the service quality of the physical network. The digital twin can be constructed from multiple components, where each component contributes to the modeling of the network. One of these components is capacity planning, which is the focus of this study.

The goal of a dynamic forecast model is to eventually attain a just-in-time capacity strategy. This strategy is aimed at the optimization of inventory usage to enhance efficiency and minimize costs. The achievement of optimal resource allocation and capacity management hinges upon the ability to provide timely and adequate resources and to adapt to inconsistent demands. In this context, the availability of precise demand forecasts is a critical prerequisite.

Secondly, this study can provide insights into the manually conducted predictions. The quarterly calculations, performed in Excel, involve data from sources beyond historical throughput to estimate future demand. These calculations consider numerous factors and at times, the procedures to derive the predicted numbers lack clear explanations. For that reason, historical data will be assessed and the knowledge gained will be leveraged to design a prediction framework. The results of this model can then be employed to validate the manually conducted predictions and extract perspectives from historical time series data.

Thirdly, risk calculations will be performed to illustrate the demand expectations with a specific level of risk, derived from statistical theories. This evaluation can help explore potential cost-benefit tradeoffs, that concern the network and its redundancy limits. It can assist in making informed decisions for intelligent purchases and investments in the network.

2.5. Implementation

To ensure the applicability of the model for future work, the implementation will be carried out in a format consistent with business perspectives. The designed model will be integrated into the programming platform DataIku [10], which is used by the company. DataIku is a comprehensive data science platform that is applicable for data science and machine learning applications. It provides a range of tools and features to streamline the data preparation, model, and deployment processes. In the context of this research, DataIku provides a visual interface seamlessly integrated with Python. This integration allows users to construct, train and assess machine learning models using Python code. This interface proves especially valuable, this creates a strong alignment between data science and strategic business objectives.

A flow, or digital environment, will be created to integrate all the steps of the prediction framework. The objective is to establish a real-time model that generates a daily forecast and visualizes these results in a dashboard. A pipeline will be established that retrieves measured data from a database and incorporates it into the flow. This data consists of daily throughput data from the network and will serve as input for the prediction framework. The data will undergo several steps of processing to acquire the appropriate format for use as input in the model. The results of the model will be computed daily and the results of this forecast can be visualized in a dashboard. Additionally, the results will be exported to another cloud observability platform used by the company to monitor real-time network throughput.

3 Literature study

This chapter will delve into the literature utilized to acquire comprehensive background knowledge on capacity planning to provide a high QoS and to do dimension core networks. Moreover, the applicability of current literature to the network traffic and models that can be leveraged for prediction will be explored. It serves as a review that examines diverse research topics and their relevance to the objective of this study.

3.1. Capacity planning

Capacity planning is pivotal to ensure that telecom providers meet the bandwidth requirements guaranteed to customers. This process spans various domains within telecom networks, which include mobile radio networks, internet routers, fiber optic connections between different locations and server capacity for specific services such as DNS and CDN. The digital revolution and the rise of the internet have transformed capacity planning into a complex and dynamic process, which covers not only voice but also data, video and other services delivered over the network. Strategic decision-making is crucial for service providers, who face the challenges of when to expand capacity, the required bandwidth for the expansions and the suitable equipment to ensure compliance with the committed QoS [11].

Various telecom services require unique transport networks due to their distinct characteristics. Consequently, capacity planning challenges come with specific objectives and constraints, which allow for the application of diverse methodologies. For instance, a case study conducted with an undisclosed prominent provider in Mexico demonstrated the effective use of inventory control techniques to implement a capacity expansion plan [11]. Similar to inventory management techniques in manufacturing, this approach demonstrated its effectiveness in the optimization of operations. While it does not address extreme value prediction, it offers insights into potential cross-disciplinary engineering-based capacity planning strategies. Additionally, data center infrastructures require specialized traffic control techniques [12]. A review explores data center network architecture, traffic properties and objectives, and discusses challenges such as prioritization, load balancing and traffic scheduling. This highlights the importance of the employment of a combination of various traffic control techniques across different network layers to improve performance metrics.

In the domain of capacity planning for the ZARA core locations, accurate traffic predictions based on historical data that includes extreme peaks, are essential. The complexity of capacity planning is significantly influenced by the transmission technology, user demand and planning horizon. If the forecasted demand exceeds available capacity, it also involves decisions on the timing and quantity of new equipment needed to meet additional demand. This process is complicated by the user demand, especially in terms of throughput and latency for applications, that continues to rise.

Capacity expansions in telecom networks follow extended time cycles. Telecom operators commonly need six months to integrate for instance a 4G and 5G layer and a span of two years for the construction of a new base station [13]. Additionally, Capital Expenditure (CapEx) investments have to be justified and proactive planning becomes essential. Therefore, the decision-making process requires accurate predictions of future network performance and must take into account various scenarios about traffic growth and capacity expansions. Overestimation of traffic growth should be avoided as it leads to high costs. Moreover, equipment may remain underemployed, an undesirable outcome given the finite operational lifespan of hardware components.

Each planning problem in telecom networks depends on multiple factors, such as the number of users, user speed and the overbooking factor [14]. In a study for a hypothetical operator in the Netherlands, a calculation for this demand in Mbps per km^2 is given. This can be obtained by the population density, estimated smart device users and market share. The user speed relies on desired bit rates, while the overbooking factor assumes that not all users will concurrently use maximum bandwidth. These parameters are intertwined with the network performance and reflect the intricate nature to predict future demand for capacity management. Accurate predictions are essential to ensure network resilience for critical failures and various traffic growth scenarios.

Big data analytics in telecom offers insights for capacity planning strategies [15]. Proactive network optimization, particularly in 5G networks, is critical to address rapid traffic growth and demanding service requirements [16]. The integration of big data analysis, cloud computing and machine learning techniques to accurately forecast traffic demand and manage uncertainties, is a valuable approach. A different paper proposed an adaptive capacity and frequency optimization method for wireless backhaul networks using time series forecasting [17]. It emphasizes the shift from reactive to proactive intelligent transport planning for mobile networks, as dynamic resource optimization proves to be an effective approach that can be used in capacity planning.

The existing research elaborates on various methodologies to approach capacity planning. However, it does not address the applicability for large-capacity fixed transport core networks and the management of extreme values, which can have a significant effect on the network performance. This gap highlights the need for prediction approaches that address extreme values in network traffic for capacity planning purposes.

This literature review aims to dissect and understand the complexities of network traffic. It will focus on prediction models for extreme values in network traffic, which enhances capacity planning in large-capacity fixed transport core networks. Prediction approaches will be reviewed to address extreme values, for efficient network design and capacity planning.

3.2. Network traffic predictions

An important part of capacity planning is to forecast future network traffic demand. Different methodologies for the prediction of network traffic have been researched in the past few years. An extensive overview of these approaches is provided in the literature [18], however, there are still various gaps for improvement as concluded from the current literature. The most relevant gaps are described below.

While many research papers underscore the importance of network traffic load prediction, there is a gap in how these forecast models can be applied in real-world scenarios [18]. This limited practical application does not address optimization challenges, such as quality of service improvements, comprehensive network control and more. There is a clear necessity for new solutions to be put into practical applications and not only assessed in comparison to existing methods based on relative prediction errors. Then the advantages of a particular model can be offered over another when deployed in real-time production environments.

Furthermore, the existing literature on efficient prediction of network traffic for real-time applications [19] primarily focuses on general network traffic prediction. This research therefore delves into the specific aspect of the prediction for extreme network traffic values, a critical area that is often less explored. Methodology to improve extreme value prediction will be explored, which contributes to the knowledge base of network capacity planning. It prompts a closer examination of methods and their practical implementations. Moreover, the real-world implementation for the transport core network of KPN aims to bridge the gap, to offer actionable insights for the telecom company in the capacity management of their network. Therefore, in this research, time series prediction methods are surveyed to determine the approaches suitable for extreme value network traffic prediction of the fixed core network. This will help improve the unique challenge of timely dimensioning the core network to obtain a stable network.

3.2.1. Time series prediction

A common approach to realize network traffic prediction is the employment of time series analysis. The aim is to identify patterns in historical data to incorporate them into the prediction model of future values. Time series prediction can be described in several steps [20], as illustrated in Figure 3.1:

1. **Data exploration and preprocessing:** Firstly the time series dataset is explored to understand the characteristics of the data. Preprocessing steps are also done to ensure that the format of the data is suitable for building the model. This involves handling potential missing values or inaccurate outliers.
2. **Training-test split:** Secondly, the time series is split up into a training set and a test set. The training set is the sequence of data points before the forecast horizon which is used to build the model. The test set involves the data to evaluate the performance of the prediction and is defined by the forecast horizon.
3. **Model selection and parameter tuning:** The third step involves the model selection based on the characteristics of the data. The set of parameters is selected by manual tuning or a search technique. The input of the model consists of the training set, which is divided into samples for training and validation, and a predefined parameter set. The algorithm tries to minimize the predictive error by identifying the most optimal parameters. The prediction model is built with this set of parameters and the training set is fitted.
4. **Prediction and error calculation:** In this step, a multistep prediction is performed for the period of the forecast horizon. The prediction error can be calculated using these values.
5. **Model evaluation:** In the fifth step, the model can be employed to compare the forecast results with the test set, to measure the accuracy of the model according to metrics like Mean Squared Error (MSE) or Mean Absolute Percentage Error (MAPE). For some algorithms, this is also known as the backtest.
6. **Model deployment:** The final step is to choose the most accurate model based on the evaluation results. The selected model can be deployed to forecast the values for future periods with the specified forecast horizon.

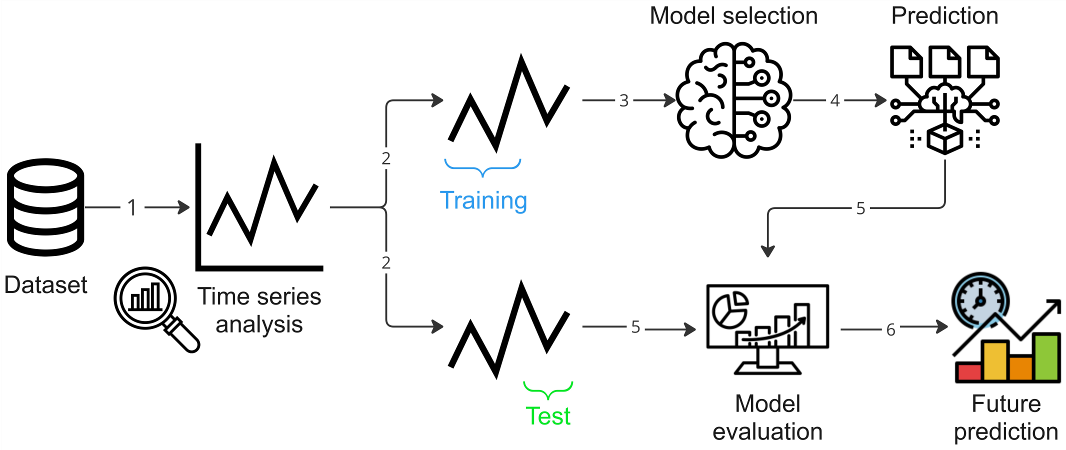


Figure 3.1: The process of a time series prediction model.

3.2.2. State of the art

Research on time series prediction in the context of capacity planning has extensively explored various statistical models. As the field evolves, machine learning models have demonstrated enhanced predictive capabilities, which play a crucial role in shaping adaptive and accurate forecasts for network traffic prediction. Both statistical and machine learning models are examined and the models suitable for addressing the challenges of network traffic prediction will be discussed.

Statistical models

Statistical techniques can be employed to characterize network traffic [21]. The ARIMA model is a class of statistical models that can be used to analyze and forecast time series data [22]. It stands for Auto-Regressive (AR) Integrated (I) Moving Average (MA). It is based on autocorrelation, which relates to the correlation between data points in consecutive time intervals [23]. ARIMA consists of the AR model, which uses the lags of previous values, the I for differencing and the MA model, to determine the error terms. As most time series models in real-life practice are not stationary, the time series have to be made stationary to apply regression techniques to the time-dependent variables to make predictions. This is performed by differencing the values of the time series by a specific order, which results in the ARIMA model. The model uses three parameters ARIMA(p,d,q) to produce a forecast [24]. Seasonal ARIMA (SARIMA) is an extension of the ARIMA model, designed to handle time series data with seasonal patterns. It incorporates seasonal differencing to address the issue of seasonality in the data and introduces the seasonal components; ARIMA(p,d,q)(P,D,Q)[m]. More on these parameters can be read in Appendix B.1.

As an example, additive and multiplicative decomposition models and the Auto-Regressive Integrated Moving Average (ARIMA) model are explored to forecast network traffic [21]. The data employed is daily usage in four days and the results show that ARIMA provides the highest accuracy. A similar study employed an ARIMA model to predict long-term evolution (LTE) throughput on weekdays, which demonstrates accurate forecasts with the approach [25]. Regression ARIMA models have established themselves as a prevalent choice to model linear time series data within a classical statistical framework [26] [27]. ARIMA models are built under the presumption of stationary data. In applications where various fluctuations are expected in the data, a decreased model performance can be expected. Nevertheless, employing transformations to remove the non-stationarity in the input data, can partially address this limitation [28].
However, ARIMA models fall short in capturing nonlinear patterns within the data as they do not support time series with seasonal data [29]. This inability makes them ill-suited for modeling intricate dynamic real-world scenarios. Therefore, SARIMA was introduced as an alternative method, for forecasting Universal Mobile Telecommunication System (UMTS) data traffic [30]. This method does not only take into account the trend elements, p, d and q but also the seasonal components; P, D, Q and m. In this case, a seasonal difference can be performed. SARIMA has been extended with the use of exogenous (X) variables to enhance accuracy by reducing error values. SARIMAX can be employed with exogenous factors to predict the long-term performance of the electricity sector [31]. This demonstrates the effectiveness of SARIMAX compared to the simpler ARIMA models.

Another study suggested a more complex model, which combines SARIMA components and the nonlinear Generalized Auto-Regressive Conditionally Heteroscedastic (GARCH) model [32]. This model addresses long-term dependencies to characterize and predict mobile communication network traffic. Moreover, a different paper employed a multiplicative SARIMA and Holt-Winters model for traffic prediction, to examine short-term prediction [33]. From the existing literature, it can be inferred that the integration of multiple models enhances robustness and improves the capability to uniformly capture various patterns within a time series. Consequently, the adoption of hybrid models or the combination of several models has become a commonplace practice. Despite this, it is noteworthy that ARIMA continues to be a standard among baseline models. SARIMA serves as a benchmark method for seasonal data, which makes it a potential fit for the analysis of network traffic data.

Machine learning models

Significant advancements have occurred in the research landscape of time series prediction that use machine learning models. Particularly, there has been a focus on addressing the complexities of dynamic and evolving datasets. The supervised Decision Tree and AdaBoost regressor learning algorithms neglect long-term dependencies and fail to predict non-linear traffic behavior [34]. Recurrent neural networks (RNN) on the other hand, can forecast traffic samples by employing a variable sliding window algorithm. This algorithm uses the traffic data from the initial data points to predict future traffic patterns, based on the training information. Given the limited size of the sliding window, past predictions are reintroduced to predict future samples.

Recent studies have explored more advanced techniques like Long Short-Term Memory (LSTM) networks and other deep learning architectures [35] [36] [37]. RNNs based on LSTM units, are designed to overcome the inherent gradient disappearance issue in RNN models [38]. It demonstrates an effective capability in non-linear time-series modeling. Furthermore, Gated Recurrent Units (GRU) were proposed as a forecast approach, next to RNN and LSTM networks [39]. GRUs are a variation of RNNs, that employ gating mechanisms to control the flow of information within their cells. The gates allow the model to selectively update its memory over time. The results of the models on GÉANT and Abilene datasets were promising for volume, packet protocol and distribution predictions. Another paper focuses on accurate one-hour-ahead forecasts of telecom activity data from a telecom provider in Vietnam, using LSTM and GRU networks [40]. The results have shown that deep neural networks hold the potential to analyze time series data from a telecom network. Moreover, the LSTM and GRU models proved to be the most reliable for all performance criteria compared to the other methods tested, such as the Artificial Neural Network (ANN) model. The models aim to capture the temporal dependencies and patterns inherent in time series data.

A comparison between the two models was executed, to study the ability of the RNN architectures to memorize sequences of varying complexity [41]. The learning rate and the number of units per layer are identified as the most crucial hyperparameters. Overall, GRUs demonstrate superior performance on low complexity sequences, while LSTMs excel on high complexity sequences. Considering more extended forecast horizons, a single LSTM network may encounter challenges in the training process to reach the most effective set of weights and parameters minimizing prediction errors. However, stacking multiple LSTM layers can enhance the capacity of the model to capture complex patterns when dealing with a larger amount of historical data.

Additionally, efforts have been directed toward incorporating exogenous variables and enhancing model intractability. One of the models that can be improved by exogenous information is DeepAR [42] [43]. DeepAR is a machine learning method designed to generate probabilistic forecasts [44]. The RNN architecture produces probabilistic predictions using Monte Carlo samples, which enables the computation of consistent quantile estimates in the prediction horizon. The cells used for modeling the RNN structure are LSTM. DeepAR distinguishes itself from classical forecasting approaches in two key aspects. Firstly, the model learns seasonal patterns and covariate dependencies across multiple time series automatically, minimizing the need for supplying covariates. It can model various types of seasonal patterns. Secondly, the approach excels in providing forecasts for series with limited historical data, as it leverages insights from similar items. This multivariate capability addresses scenarios where classical univariate forecasting methods fall short. A DeepAR model that considers the non-linear and non-stationary characteristics of network traffic was introduced for base cell station traffic [45]. The proposed DeepAR model incorporates artificial feature sequences based on local moving averages (LMA) to enhance the long-term prediction performance of multi-cell network traffic. It outperforms other methods, such as ARIMA, XGBoost and LSTM in terms of accuracy and reliability of predictions.

3.3. Relevant prediction models

The time series contains extreme values that are important to forecast. Consequently, models that can incorporate the use of exogenous variables to improve the prediction of these extremes will be chosen for the design of the prediction framework. A statistical model and a machine learning model will be compared and tested on their applicability to the dataset of the fixed core network.

The statistical model that will be employed is SARIMA(X), which is the extension of an ARIMA model. Further details on SARIMA(X) and the model parameters are elaborated upon in Appendix B.1.

When choosing the machine learning model, it is important to consider the following. The employment of complex deep learning models with numerous internal parameters becomes unfeasible when working with short time series. When a limited number of data points is available, less than a hundred data samples for instance, the series may not be suitable for the complexity of a deep learning model. This is particularly in situations that involve data aggregation at a lower granularity, such as monthly data, as it may not effectively capture meaningful patterns within the data [46]. As the prediction framework will be designed for the forecast of daily data, this should not be an implication. Moreover, the interpretation of machine learning models, particularly deep learning models, can be a challenge due to their inherent complexity. A lack of transparency in the decision-making process is unwanted. The trade-off between model performance and interpretability is a crucial consideration.

The machine learning model chosen for this research is DeepAR. This model has shown that it delivers robust results. The challenge with DeepAR is that the algorithm employs deep learning techniques, which is less interpretable than traditional statistical methods. However, the model is better at the recognition of complex patterns in the data. In addition, various time series can be correlated to enhance model training. This possibility shows high potential when the correlation between the usage of different domains is analyzed and used for a dynamic forecasting model.

While ARIMA models each time series independently and predicts univariate time series, DeepAR learns from multiple time series simultaneously [44] and can create a multivariate prediction model. This makes it capable of using information from similar items to make predictions, even when individual time series have limited historical data. This ability to see and consider patterns or similarities between different related time series is a distinct advantage of DeepAR over a traditional method such as ARIMA. Although this offers predictive advantages, this aspect is not explored currently. This research focuses on using individual time series as inputs to the forecasting models. This forms the foundation of a fundamental analysis of time series forecasting for the two network components. Considering future work, the significance of employing this method becomes especially valuable when working with multiple network streams.

Extreme events or outliers in the data can adversely affect prediction models. In time series, extreme events are typically characterized by extremely small or large values, that occur at random [47]. For the prediction of time series with extreme values, the SARIMA models proved impractical for time series data that contained extreme values [48] [49]. Moreover, deep learning approaches have to be enhanced to focus on modeling extreme events more accurately [50]. A novel approach, using extreme value theory and the incorporation of a Memory Network, was introduced to improve network capacity planning for extreme values. Therefore, to tackle the challenge of extreme events, extreme value theory can be considered to model extreme values.

3.3.1. Extreme Value Theory

As previously described, challenges arise for time series models that contain extreme events. Their existence can significantly impact the overall effectiveness of time series models. Extreme Value Theory or Analysis (EVA) presents a potential solution to address these challenges. It is a statistical theory that uses the extreme values of a time series and its tail distributions to provide insights into the occurrence and impact of extreme events. It focuses on significant deviations from the median within probability distributions. The objective is to forecast the likelihood of extreme events, that lie outside the available range of data [51].

It is widely applicable in diverse domains [52], which includes finance, hydrology, road traffic prediction and structural engineering. Examples of extreme events are financial market crashes or rare weather phenomena, such as floods. For instance, the Netherlands has applied EVA for the challenges and risks associated with floods, which is relevant in a country that is susceptible to sea level extremes. Applications of EVA are specifically risk management, Value-at-Risk (VaR) estimation and insurance [53]. These three topics can be described as follows:

- **Risk management:** It serves as a method to assess tail risk, which is the probability of extreme events occurring. EVA models the distribution of extreme values and facilitates the quantification of the likelihood of infrequent yet impactful events. This is a critical aspect in devising effective risk mitigation strategies. Additionally, more precise modeling of extreme events becomes feasible, which enables stakeholders to incorporate tail risk measures when making investment choices in the network.
- **VaR estimation:** VaR is a risk metric that calculates the expected highest potential loss within a specified time frame and confidence level. EVA offers a reliable approach to VaR estimation by modeling the tail distribution of, for instance, financial returns. This method enables the estimation of the potential losses in extreme situations and provides insights into worst-case downside risks

associated with an investment [54].

- **Insurance:** EVA is useful in the insurance and reinsurance industries for modeling significant losses arising from catastrophic natural events, such as hurricanes and earthquakes. Through the examination of historical data, EVA aids insurance companies in estimating the tail risk linked to these events. This enables them to set policy prices in alignment with potential risks. Reinsurance companies similarly employ the theory to evaluate potential losses that can arise from catastrophic events and their reinsurance requirements.

EVA has seen limited applications in the telecom sector, particularly in the domain of network time series prediction. Nonetheless, the researches that employ EVA for telecom data are elaborated on in the next described papers.
One study applies EVA to teletraffic data to investigate the heavy-tailed nature of internet traffic [55], in this case the file length requested to a server. The analysis concludes that the requested file sizes follow a long-tailed distribution, similar to a Pareto distribution, which indicates that the requests are rare but significant. This shows that internet behavior can be modeled with EVA for the right tail of the distribution.
Moreover, EVA is introduced to predict telecommunication quality deterioration with a small amount of known data [56]. The analysis of throughput tail distributions outperforms other methods to predict the tail distribution of unknown data. It is more accurate than empirical or Log-normal distributions and enables cost-effective forecasting of significant events with reduced measurement and storage requirements.
Another paper focused on wireless network traffic analysis with extreme value theory [57]. EVA effectively characterizes traffic data and shows a lower average deviation compared to other distribution models such as Exponential and Log-normal. This demonstrates that EVA can be employed to estimate extreme behavior in random processes, as it provides more accurate predictions and reduces computational overhead.

The application of Extreme Value Theory in the analysis of extreme events has demonstrated its effectiveness in this sector and especially in other disciplinary fields. The success of EVA is shown as it outperforms other methods for network traffic data. This underscores its potential as a robust tool for the development of accurate prediction models. For this reason, further exploration into the capabilities of EVA will be described in Section 5.2.2.

4 Data analysis

This chapter will provide detailed information about the service layer of the network and available network traffic data. Additionally, the chosen data to be used as input for the model will be visualized and analyzed to gain deeper insights into usage behavior and traffic peaks in the network data. The relationship between events and peaks will be investigated, which enables the creation of a predictive model based on this knowledge.

4.1. Data selection

The downstream throughput of the network is measured by data collection instruments. These are installed on each link within the network, to monitor the traffic load. The data measurement pipeline stores only the date of the maximum throughput and does not store the timestamp correlated to the daily peak. This prevents any analysis of the time aspect of the maximum throughput besides the date of the measured peak.

The throughput of the transport core, which originates from the service layer, is measured at two levels in the network. These levels are illustrated in Figure 4.1. The first level involves the throughput of the connections between the specific provided services and the domains that facilitate these services. These include external applications from content vendors. The second level includes the links to the domains accessible from the transport core. The throughput measured here consists of the total aggregated throughput of all different services for every service domain.

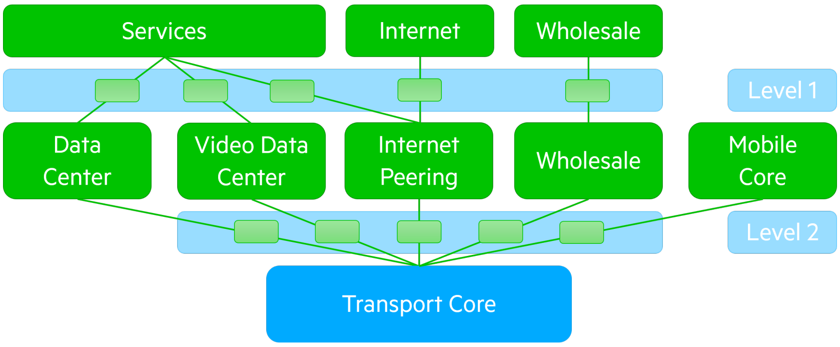


Figure 4.1: The levels where data is measured in the service layer.

4.1.1. Data granularity

Traffic prediction relies significantly on the time granularity of the dataset. This influences two main aspects to choose a suitable prediction method. Firstly, the length of the available time series plays a pivotal role in the forecast horizon possible. The forecast horizon defines the length for which the prediction is calculated. It is important that during the training of a prediction model, the number of observations must surpass the number of model parameters [58]. Secondly, the aggregation of the data can influence the performance of the prediction models. Aggregation to a lower granularity can eliminate variations in the data and result in fewer data points being available for training the model, which affects the predictability of the data. Hence, it is important to establish a granularity aligned with the intended insights to be derived from the forecast results.

The data on the links from level 2 was chosen for this research. A significant factor in this choice was the data granularity. Data granularity refers to the degree of detail present in the data structure. To decide on the level of granularity, it is important to assess the data quality. The total throughput of the measured datasets of level 1 and level 2 should be identical as they involve the same traffic. However, measurement errors in the instruments can lead to variations in the data. The selection from which level the measured data should be taken involved a trade-off. Data from level 1 provides more specific insights into user behavior and service usage, but it is less accurate in terms of measurements. This is because some traffic is not measured directly on the network links, as these numbers are provided by external content vendors and thus the accuracy of the data can not be validated. On the other hand, data from level 2 is more accurate as the traffic on these links is measured by instruments of the network. However, it provides a higher-level granularity of service usage which offers fewer details about user behavior. Nevertheless, the data of level 1 obtained numerous absent data values and as some of these measurements were not all directly measured by the network itself this can not be retrieved, so this dataset was excluded from the research. As data accuracy is crucial for a prediction model, more accurate data is preferable. Hence, the measured data on level 2 is chosen for this research.

From January 2017 until March 2020, the stored data only includes the maximum traffic reached within one day per week. This data represents the weekly historical data measured by the data collection instruments. Since March 2020, the daily maximum throughput was also stored. This period coincided with the onset of the COVID-19 pandemic in the Netherlands, which accelerated the digital transformation for companies. The usage of network traffic changed and the throughput increased drastically, as people were suddenly forced to work from home [59]. Subsequently, this caused a shift in work patterns as an increased number of people adopted a hybrid work model, that alternates between working from home and in the office. This hybrid way has persisted in current society. Accordingly, the data collected from the start of the pandemic until the present is more indicative of current and future usage compared to the traffic data before the pandemic. In the forecast process, an underlying assumption is that historical conditions resemble future conditions. Furthermore, the use of daily data is more suitable for the design of the prediction model, as it provides more detailed information on specific situations of high network peaks compared to weekly data. This is particularly important in light of the future objective to create a dynamic prediction model that considers extreme values. Therefore, the dataset chosen consists of the daily network traffic peak data spanning from March 2020.

4.1.2. The service domains

As described, the input data for the design of the prediction model was acquired from level 2; the links that connect each service domain and the transport core. The goal is that the model will be employed to predict the throughput of all service domains, to gain insights into the capacity needed on the transport core links. It was chosen not to employ a time series that represents the maximum total throughput of the transport core. Firstly, because the five time series can not be accurately aggregated to obtain one time series for the throughput. As stated before, the daily peaks do not contain the time at which the maximum daily value occurred. Therefore, aggregation of the five domains is not possible. This would give a time series that contains all maximum throughput values per day, even though they might not occur at the same time, which is not a realistic representation of the maximum throughput. If for instance, multiple users stream live content, fewer people make use of VOD content. Secondly, an aggregated time series would give a more high-level view of user behavior, which does not provide enough granularity for the objective of this research. Hence, an analysis of one or more of these time series will be focused on for the analysis.

As the prediction model had to be created for the current network infrastructure, the time series have been considered as separate components. Estimation of network traffic uses past measurements of identical links in the network to predict the throughput in the future.
All service domains show a recurrent pattern throughout the years. This pattern will be described further in Section 4.2.3. The throughput of each domain has its characteristics:

- **Data Center**: The average throughput of the Data Center remains stable, except for the infrastructure change as previously described. The reallocation of the traffic load is apparent in the measured data and caused the sudden decrease in throughput.

- **Video Data Center**: The traffic load is reallocated to this new domain. After this change, the average throughput shows to maintain the same level of traffic. This could be linked to the fact that the CDNs efficiently employ multicast transmission, which ensures that extreme peaks are not likely to occur. Also, the overall throughput is not that high as the most popular content is not visible in this data, as this is decentralized to the access level.
- **Internet Peering**: In contrast, the time series of Internet Peering exhibits a steady growth in trend and extreme peaks compared to the average throughput. The average throughput is the highest of all five service domains.
- **Wholesale**: The traffic of the wholesale links remains generally stable compared to consumer network traffic, due to the services provided. It involves consistent data patterns and is less subject to unpredictable user behavior. A slight increase in trend is visible and contains a few noticeable peaks.
- **Mobile Core**: The daily Mobile Core measurements began in April 2021, which resulted in a shorter time series. Additionally, the amplitude of the Mobile Core throughput is smaller than that of Internet Peering, which results in a smaller load on the transport core network. Nevertheless, this time series provides valuable insights into customer streaming behavior and makes it an important aspect to take into account.

The design of the prediction model for the transport core is based on the historical time series data of one service domain. This is to create a comprehensive analysis of this time series. The choice between the time series was based on the comparison of the different domains and their functionalities. The conclusion was that the Internet Peering time series is the most valuable to analyze. This traffic has the highest trend increase and shows the most deviations from the average throughput. The traffic peaks become more extreme over time, thus this time series can be considered the most challenging to develop an accurate model for. A model that can predict the throughput based on the characteristics of the Internet Peering time series, should be able to predict the demand for the other service domains. Moreover, the traffic data of Mobile Core has also been considered for time series analysis. Although the traffic load is relatively smaller, this data offers valuable insights into user behavior.
For the design of a reliable prediction framework, the time series of the Data Center and Video Data Center links are not suitable due to the sudden change in traffic behavior. Due to the alterations in network behavior by the infrastructure adjustments in these domains, significant challenges to traffic prediction methods arise in pattern recognition. Additionally, if only the part of the time series after the change is taken into account, as this is the design of the current network, the time series are too short to perform predictions with a forecast horizon of one year in advance. Furthermore, the wholesale traffic is not considered for the time series analysis as it does not offer additional insights into user behavior, given the nature of the service provided in that domain. For that reason, the time series of Internet Peering and the Mobile Core are considered for subsequent time series analysis and design steps for the model.

The prediction model will be constructed with the Internet Peering time series and insights derived from the Mobile Core time series. Initial evaluation will focus on the Internet Peering dataset, followed by its application to other time series. For shorter time series, necessary model adjustments have to be made to enable predictions of the different domains. For the Data Center and the Video Data Center, the model could be trained with a shorter forecast horizon until the time series taken after the infrastructure change, extends to a sufficient length to be predicted for one year in advance.

Measurement pipeline

The selection of the time series for the prediction model showed that the current measured data has its limitations. As explained in Section 2.2, the data is collected at five-minute intervals, which results in a daily traffic profile of the network throughput. However, only the maximum value per day thus one throughput value is stored and the daily traffic profile is discarded in the current data measurement pipeline. This information holds potential value to enhance the network analysis in several ways. Firstly, the daily throughput profile offers a more detailed perspective on network performance compared to solely the peak value. It enables the creation of a more comprehensive dataset that incorporates fluctuations and trends throughout the day, rather than solely the daily throughput. A daily traffic profile offers insights into user behavior and usage patterns over time. The distribution of traffic throughout the day can help to accommodate fluctuations in demand and correlate the usage to events. Moreover, a daily traffic profile could be created for every day of the week. The throughput flow varies throughout the day but also per day of the week. This information can be invaluable to optimize resource allocation, to obtain a dynamic prediction model in the future. This can lead to cost savings and improved network efficiency.

4.2. Data preprocessing

The collected data of the Internet Peering and Mobile Core domains is analyzed and processed for further examination. The other domains will be predicted with the same framework after the final design has been created. The process involves the transformation of raw data into the desired format, addressing missing values through interpolation techniques and conduction of a time series analysis. Key components of this analysis include time series decomposition and stationarity tests, which provide a robust foundation for extracting meaningful insights from the dataset. Each step is executed to uncover patterns, trends and fluctuations within the data. This facilitates a comprehensive understanding of the network traffic dynamics.

4.2.1. Data cleaning

The obtained daily peak datasets have a few days missing in the set. These are 9 days at the end of August and 5 days at the beginning of September. These peaks are missing due to unstable measurement tooling that stopped working. Therefore the incomplete dataset was interpolated after consultation with the expert responsible for the maximum network throughput data. Table 4.1 shows the deterministic procedure followed to interpolate the missing values [60], with $Y_{dd/MM}$ representing the maximum throughput of that specific day (d) and month (M). For instance, on Friday 21/08, the assigned value is the mean of the maximum amount of traffic from the throughput on Thursday 20/08 and Saturday 22/08. As the following two weeks were missing, these throughput values could not be imputed from the mean from the throughput the days before and after but had to be chosen using another solution. Sunday 23/08, a Sunday without any event or F1 race day, uses the same throughput as the next first Sunday without a race day, which was 20/09.

Table 4.1: Interpolation calculations for missing data values

Throughput missing	Interpolation method
Friday 21/08 ($Y_{21/08}$)	$Y_{21/08} = \frac{Y_{20/08}+Y_{22/08}}{2}$
Sunday 23/08 ($Y_{23/08}$)	$Y_{23/08}$ = $Y_{20/09}$
Monday 24/08 ($Y_{24/08}$) to Friday 28/08 ($Y_{28/08}$)	$Y_{24/08}$ to $Y_{28/08}$ = $Y_{17/08}$ to $Y_{21/08}$
Saturday 29/08 ($Y_{29/08}$)	$Y_{29/08}$ = $Y_{22/08}$
Monday 31/08 ($Y_{31/08}$) to Friday 04/09 ($Y_{04/09}$)	$Y_{31/08}$ to $Y_{04/09}$ = $Y_{07/09}$ to $Y_{11/09}$

4.2.2. Relative difference dataset

To isolate the extreme events of the daily maximum throughput, a new dataset is generated from the historical time series that consists of the deviation between the average throughput and the daily peaks. This approach has been adopted to improve the extraction of the extreme values, as the average throughput varies seasonally. Due to this, a peak that occurs in the summer might be lower in absolute value than a peak that occurs in the winter, but can be significant compared to the average throughput. To achieve this, a rolling moving average of 31 days was computed for the throughput data. The moving average of 31 days is chosen as this includes the throughput of four weeks in the calculation. If there are any weekly patterns in the data, for instance, a higher throughput on Sundays and lower traffic usage throughout the week, a 31-day window can better capture these patterns. Shorter windows with like 7 days or 14 days, would introduce more daily fluctuations and short-term trends and would be more sensitive to abrupt changes.

The moving average of 31 days represents the average throughput over a period one month and has been calculated as shown in Equation 4.1. Here k is the size of the sliding window and y_i is the ith data point in the set of data points $y_1, y_2, ..., y_n$ [61] [62]. The sliding window is set on 31 days, thus the data from the previous 30 days is needed in addition to the value of the current day, to determine the moving average for the current day.

$$\text{MA}_k = \frac{1}{k} \sum_{i=n-k+1}^{n} y_i \tag{4.1}$$

This results in the moving average with a window of 31 days as shown in Figure 4.2.

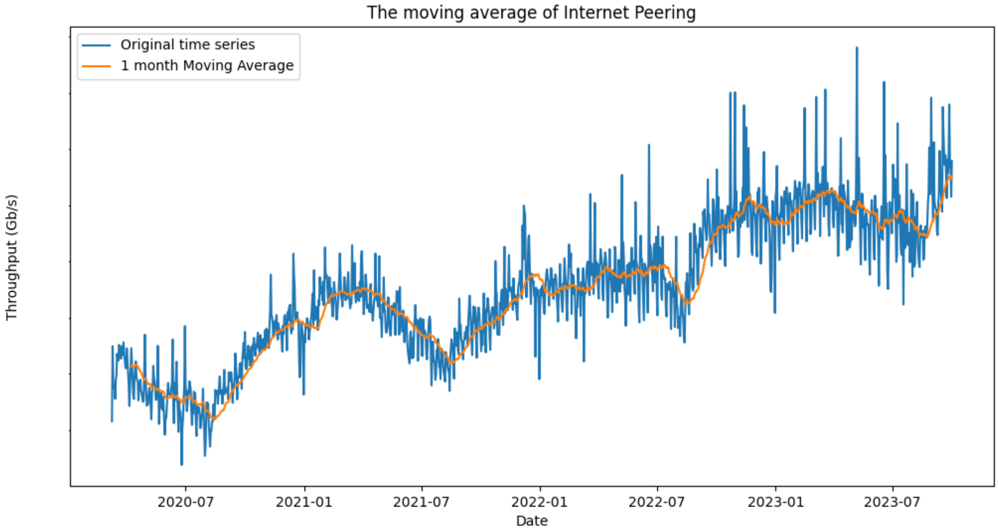


Figure 4.2: The Internet Peering time series with the moving average of 1 month.

Subsequently, for every day the difference between the daily throughput value and its corresponding 31-day moving average is calculated, as described in Equation 4.2. In this manner, the variations from the expected mean are measured. This dataset is then used for the further EVA steps.

$$\Delta_i = y_i - \text{MA}_k(i) \tag{4.2}$$

This results in a new dataset, that contains the differences between the average throughput in a month and the maximum peak per day. Figure 4.3 illustrates the dataset. It can be seen that the deviations from the average throughput increase over time and peaks are more extreme than before.

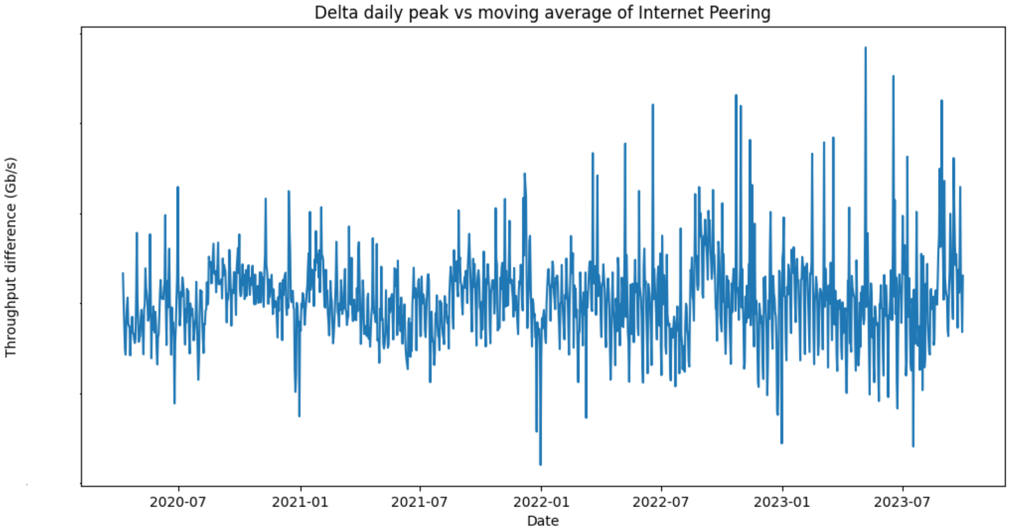


Figure 4.3: Generated dataset Δ_i of the Internet Peering throughput.

4.2.3. Time series analysis

The time series analysis has been performed on the measured data points of the two domains shown in Figure 4.4; 1302 days of Internet Peering and 898 days of Mobile Core. This data is used to assess the characteristics of the data and to extract insights from the time series. The primary emphasis in this analysis will be on the Internet Peering time series, with Mobile Core data included to provide additional insights. The designed model will be implemented for these two domains, as described in Chapter 6.

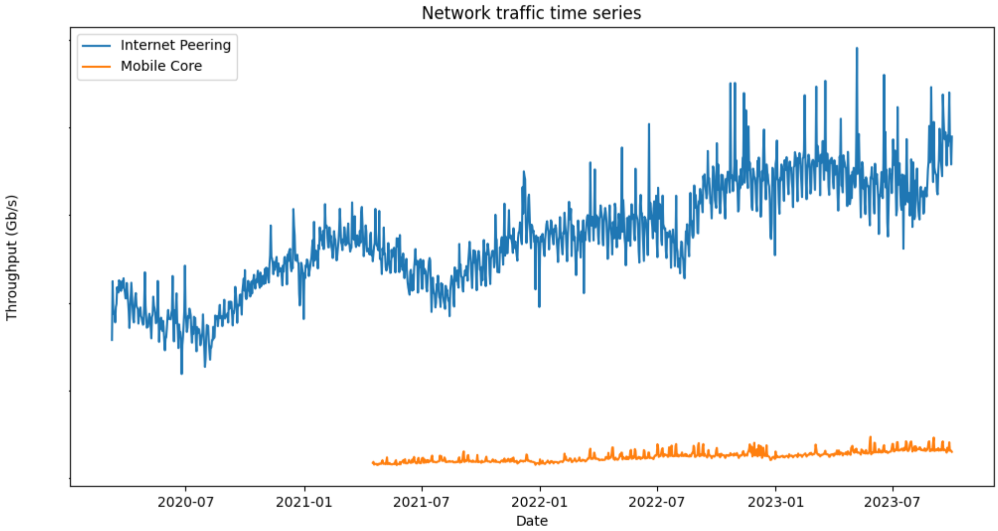


Figure 4.4: The Internet Peering and Mobile Core time series chosen for the analysis.

Time series can be decomposed into distinct characteristics, which are trend, seasonality and noise [63]. The trend of a time series is a long-term increasing, constant, or decreasing change over time. Seasonality describes whether there is a recurrent pattern within the time series over a specific period, such as weekly or monthly, which can be influenced by seasonal factors. Lastly, there is some unexplained variability in the data known as the residue. This accounts for the random fluctuations that are not accounted for by seasonality or trend and is also referred to as noise.

Various patterns can be extracted from the time series data. Empirical analysis indicates that the throughput of the time series increases over time and follows the same pattern every year. Seasonal changes likely influence this in user behavior. To help recognize these underlying patterns, time series decomposition can be performed. Classic decomposition has been performed with the programming language Python [64] to obtain the different time series components by employing the multiplicative model [65]. The idea is that the various components of the decomposition can be combined to obtain the historical time series, through a multiplicative time series model. For the multiplicative model Equation 4.3 is used:

$$Y_t = T_t * S_t * R_t \tag{4.3}$$

Here T_t represents the trend component, S_t the seasonal component and R_t the residual component.

The time series contains daily data and annual seasonality. The season length used to perform the decomposition must be 365 days to extract this seasonality. The moving average sliding window is therefore set on 365 days, which also determines the long-term trend throughout the years. The results of the decomposition following the multiplication model for the Internet Peering time series, are shown below in Figure 4.5.

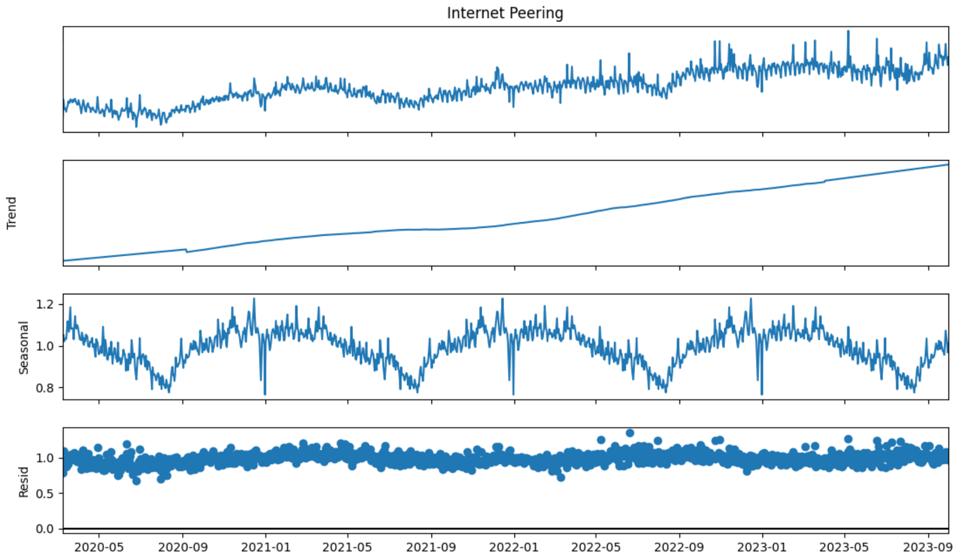


Figure 4.5: The multiplicative decomposition of the Internet Peering.

Figure 4.5 shows the decomposition by the multiplicative model. The throughput in Gb/s defines the trend component. The seasonality repeats a yearly pattern as defined in the model. The y-axes of the seasonality and the residual component of the time series model are factors instead of throughput values. These factors represent the values that should be multiplied by the trend component to obtain the historical time series. The same decomposition is performed for the Mobile Core time series and is depicted in the appendix in Figure 6.9.

The components can be extracted from the Internet Peering time series by use of a multiplicative decomposition model. Figure 4.6 provides a detailed illustration of the trend and seasonality components.

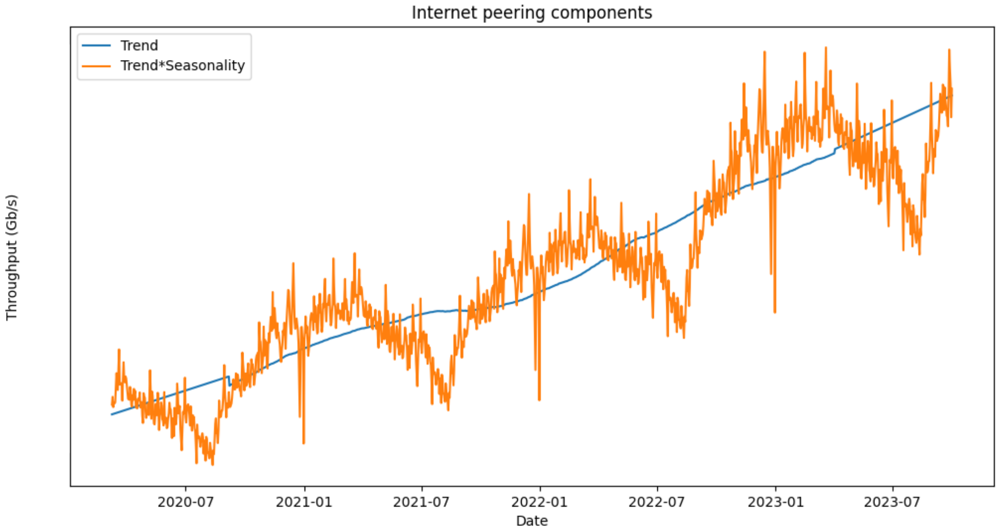


Figure 4.6: The trend and seasonality components of the time series.

The observed linear increasing trend indicates a consistent growth in traffic over the years, with a steep slope reflecting an annual increase of 40%. This upward trend aligns with the expanding demand for network services each year. Additionally, the time series exhibits an annual seasonality, as it depicts a recurrent pattern influenced by seasonal factors. The throughput experiences an upward trend from the start of the calendar year, with traffic decreasing through the summer months. The highest amount of traffic is reached during the winter months and two significant drops are evident on Christmas and New Year's Eve. Just after midnight on New Year's Day the traffic increases drastically compared to the day before and the same yearly pattern repeats itself. The magnitude of the seasonal component increases as the traffic grows.

Stationarity

Empirically it is evident that the time series is not stationary. To confirm and to determine whether the dataset is stationary, two tests have been conducted: the Augmented Dicky-Fuller test (ADF) [66] and the Kwiatkwoski-Phillips-Schmidt-Shin (KPSS) test [67]. The ADF test aims to identify the presence of a unit root in the time series:

- H_0: The Null Hypothesis assumes that the series possesses a unit root, indicating non-stationary due to its time dependent structure.
- H_1: The Alternate Hypothesis suggests the absence of a unit root, implying that the series is stationary.

When the p-value is smaller than the threshold, $p < 0.05$, the series is stationary. To further confirm this hypothesis, the KPSS test is performed as well. This test operates in the opposite direction of the ADF test, thus if $p < 0.05$ the series is non-stationary.

- H_0: The Null Hypothesis of the KPSS test assumes a stationary trend or level stationarity.
- H_1: The Alternate Hypothesis suggests the presence of a unit root thus implying the series is non-stationary.

The results of these tests can be seen in Table A.1 in Appendix A.1. The outcome of the test shows that the p-value is greater than the threshold of 0.05 and the Null Hypothesis can not be rejected, indicating that the time series is non-stationary. The results of the KPSS test yield a p-value smaller than 0.010000, which falls below the significance threshold of 0.05 and the Null Hypothesis is rejected. Consequently, this indicates that the time series is indeed non-stationary. Therefore the time series must be made stationary by differencing, which means subtracting the previous value by the current value. More on the use of this differencing is explained in Section 5.1.1.

Autocorrelation

To see whether the series is autocorrelated, autocorrelation function (ACF) tests have been performed. Autocorrelation states whether the previous values of the series, or lags, can help predict the current value. Moreover, the partial autocorrelation function (PACF) has been taken. The lags show that the traffic at a given time is influenced by the traffic in the previous period. This information is crucial to capture the temporal dependencies within the data. The results are shown and elaborated on in Appendix A.2. The autocorrelation plots help to determine the model orders for the ARIMA model. More on the employed model orders is described in Section 5.1.1.

4.3. Usage behaviour

An analysis has been conducted to obtain insights into user behavior patterns. Firstly, the dataset has been sorted to distinguish peak values that occur during weekdays and weekends. This segmentation allows for an examination of Internet usage trends per week and is depicted in Figure 4.7.

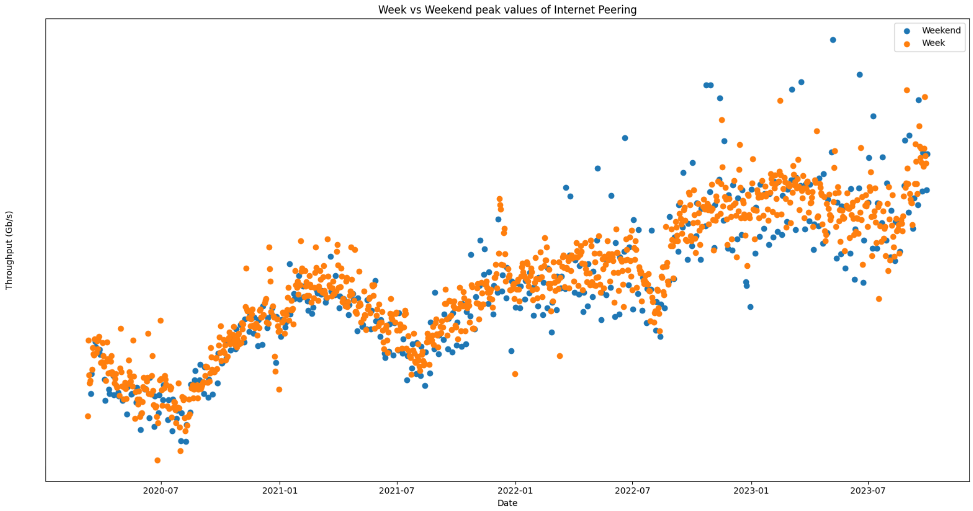


Figure 4.7: The week and weekend throughput values of Internet Peering.

It is evident that at the start of the dataset, a majority of the elevated peaks occur on weekdays. However, from the second half of 2021, there is an increase in peak values that occur during the weekends. Particularly since 2022, the most significant peaks largely emerge during the weekends. This pattern shift may be attributed to the influence of the COVID-19 pandemic, which led to cancellations or postponements of numerous popular television events. If postponed, the events occurred towards the end of 2021, which could explain the increase in peaks as F1 events often occur during the weekend.

Additionally, the data was organized and arranged based on specific days of the week. This arrangement resulted in day specific traffic data spanning 186 weeks, which shows more details on the occurrence of the daily peak values. To visualize which days generate the most Internet Peering traffic, the throughput for every specific day of the week is shown in Figure 4.8. Empirical examination reveals that a majority of the highest peaks in 2020 and the initial half of 2021 occur on Tuesdays. Tuesdays were often the days on which live press conferences on the pandemic measures occurred and could explain the higher peaks on these days. Again, towards the end of 2021, the shift in high peaks that mostly occur on the weekends instead of weekdays is evident. This can be attributed to the increasing number of high peaks that can be observed for traffic on Sundays. This trend continues and intensifies as time goes on. This confirms the increase of the network peaks on Sundays and as described in Section 2.3, they can have a significant impact on the network. Moreover, the peaks on Sunday seem to have a higher magnitude than on other days. This is an effect of the F1 race days. Overall the magnitude of all peaks increases throughout the years.

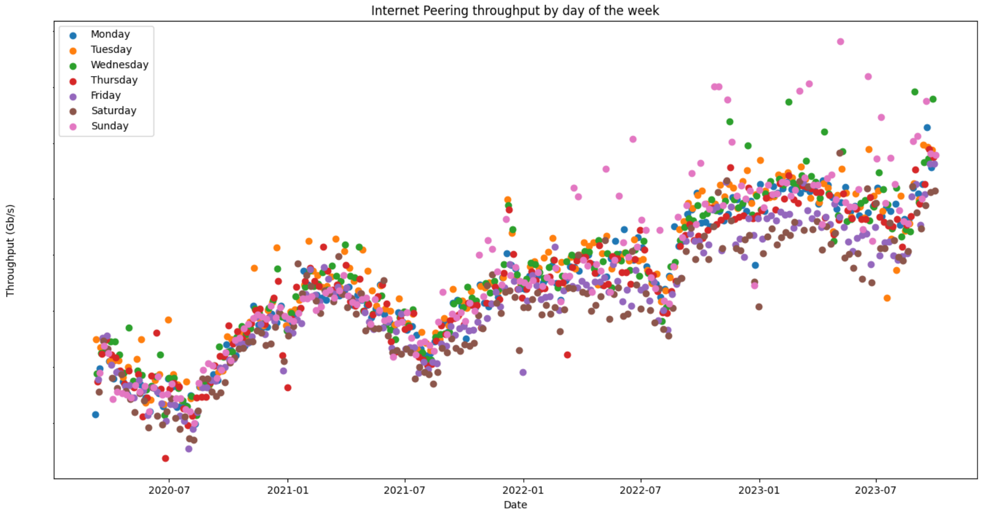


Figure 4.8: The throughput of Internet Peering for every day of the week.

Moreover, for every day of the week, the moving average with a sliding window of 52 has been calculated to obtain the average of the measured maximum throughout of the year. The results of this are illustrated in the appendix in Figure A.3 and show the days with an overall higher load on the network. This does not directly mean that the absolute highest peaks are measured on those days, but it can be expected that most extremes are on the days with the overall highest throughput average. Other days can still contain extremes, but one or two extremes in 52 weeks do not make a significant difference on the average. The moving averages are computed to see whether there is a recurrent usage pattern throughout the years. Two changes in user behavior are noticeable as the traffic on Sundays became higher than the throughput of the other days. Furthermore, the overall throughput of Wednesdays became higher than the traffic on Tuesdays and became the day with the second highest overall load.

Lastly, the average throughput per year for each particular day of the week was calculated. This provides other insights compared to the one year moving average depiction. The average throughput for every day can be seen in Figure 4.9. Moreover, the changes in increase between the annual throughput averages of every day have been calculated and reported in this figure. The percentages in 2020 are 0% as no change in increase can be calculated for the same year. The increase in average throughput from 2020 to 2021 shows an increase of around 25% for all days. In the following year when 2022 happens, an increase of 25% can be seen for traffic on Sundays. The rest of the days had an increase of less than 20%, which indicates Sundays generally exhibit a higher usage load. This substantiates the change in behavior that was previously observed.

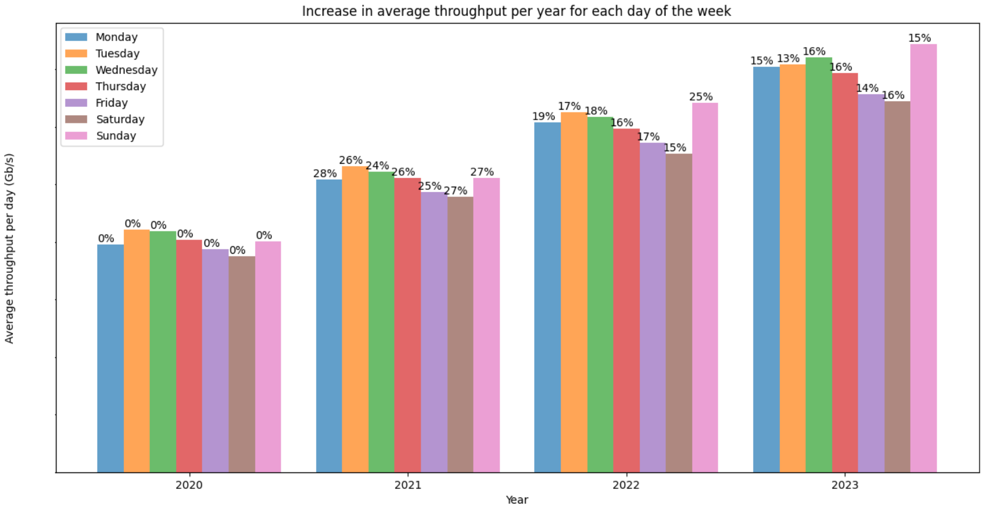


Figure 4.9: The annual increase in average throughput for every day of the week of the Internet Peering dataset.

The changes in user behavior of the network traffic data show that the throughput does not show consistent patterns during the past 3,5 years. These changes could for instance be explained by the lockdowns due to COVID-19 and result in different usage of Internet services compared to now. To understand the data, the peaks that are evident on specific days will be further analyzed to see whether a correlation can be made between high throughput values and events.

4.4. Peaks and external events

The time series data has been visualized to explore potential correlations between specific peaks and noteworthy events. Plotting the data helps identify patterns or anomalies that may coincide with significant occurrences. This visual analysis enhances the understanding of how external factors or events could impact network traffic, providing valuable insights into the dynamics of the network. The hypothesis that suggests that peaks are influenced by popular events, which could offer valuable insights for prediction models, will be further investigated.

The impact of Formula 1 events on the network is evident. As discussed in Section 2.3, F1 races can cause increased traffic peaks and are visible in the network throughput. Incorporation of this consideration into the network design helps manage substantial peaks, as demonstrated in the case of letting Viaplay provide a part of their content through multicast streaming. To examine the relationship between peaks and F1 race days, a dataset was created to document the occurrence of all F1 race days starting from March 2020. While it is acknowledged that F1 race days contribute to a high network load, this correlation was confirmed by plotting the time series of Internet Peering together with the race days. Figure 4.10 shows the days on which a race took place on the peak corresponding to that day.

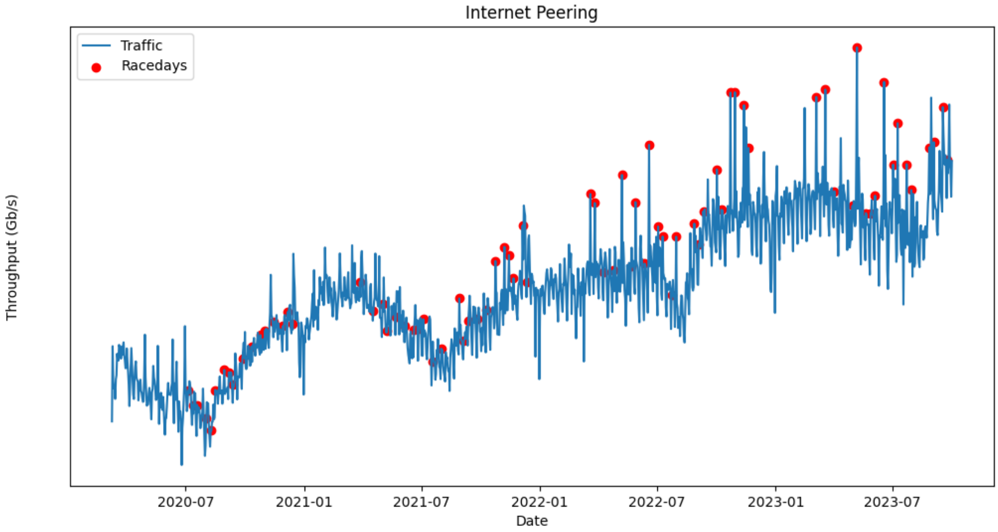


Figure 4.10: F1 race days plotted on the Internet Peering time series.

Prior to the end of 2021, it is evident that race days did not result in a substantial increase in network traffic. This might be influenced by the fact that F1 races were broadcasted by other television sports channels and are therefore less visible on the network of KPN. These peaks were the result of other significant events; for instance, some were caused by COVID-19 press conferences that were executed by the Dutch government. However, since such events are not anticipated to recur, they have not been further investigated. Additional peaks prior to 2022, can be correlated to football matches, which is the most watched sport in the Netherlands. Particularly in 2021, the postponed Euro Cup 2020 matches and Champions League matches caused high peaks in the network traffic.
This shows that the load on the network is dependent on the available content that the network offers and the viewing habits of customers. A change in offered content can shift the patterns of network traffic and cause other peaks. With new trends that will arise in the future, the highest peaks could be caused by other services. Therefore, peak analysis with events is important to understand the capacity demand.

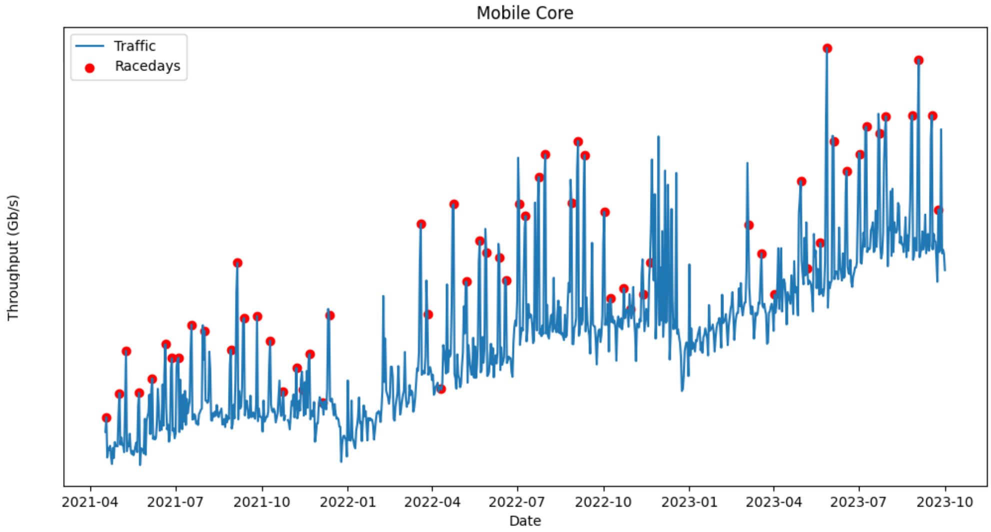


Figure 4.11: Formula 1 race days plotted on the Mobile Core time series.

A similar analysis can be conducted for the Mobile Core time series. Figure 4.11 illustrates that the race days can more frequently be associated with high peaks in the Mobile Core traffic compared to the peaks in the Internet traffic data. This underscores the importance that a correlation between time series can provide valuable extra information for the prediction. This is expected since the throughput of this domain mostly reflects the traffic generated by users streaming from their mobile devices. Both figures clearly show that a large amount of peaks, especially in the last two years can be correlated to F1 peaks, which are expected events. Taking these events into account when designing a prediction model with exogenous variables, could improve the prediction of traffic.

However, it is also apparent that not all peaks can be attributed to F1 events. Other popular (sports) events could be the cause of that. A few of them can be explained by football events. The FIFA World Cup matches in November 2022 caused significant high peaks on the network. Moreover, a significant high peak in the Mobile Core data on March 4th, 2023 can be associated with two popular sports events that took place on that day. One was a football match between Ajax and Feyenoord in the KNVB Beker tournament and the other was a World Championship 2023 Speed Skating event. Although these events can be correlated to significant peaks on the network, they were not included in the plotted figures. Since the documentation of the occurrence of these events had to be manually created, including football matches and popular live (sports) events, the decision was made to focus solely on F1 race days. This is to maintain a consistent overview of events that affect the network load, as including all events that are popular amongst users, costs a lot of time to document and missed events could result in training a model with inconsistent input information.

Most of these events are expected and are planned. However, some events are unexpected and thus not planned. On February 15th, 2023 an event by Giro 555 caused high traffic on the network. This was a fundraising campaign for an unexpected humanitarian disaster that happened a week before. Events like these are unpredictable but can occur. Therefore it is crucial to anticipate and plan for foreseeable events in advance. This proactive approach ensures that the network is adequately prepared for increased peaks. Unforeseen events of this nature can potentially lead to unexpected challenges and therefore, addressing expected issues proactively allows for better mitigation strategies.

5 Forecast methodology

This chapter describes the methodology used and the steps taken in development towards the prediction model. The initial phase of the research involves the employment of the time series data as input for ARIMA and DeepAR. This assessment showed that these models did not provide accurate predictions for the extreme values, which helps to determine further research steps to improve and obtain a substantiated forecast. Therefore, a statistical method was chosen to predict the expected throughput considering seasonality and trend, which involves the application of extreme value theory to the time series to model the risk of extreme events.

5.1. Time series models

In time series prediction, datasets are analyzed to develop a model for a specific target variable. The underlying relationship of the observations in the dataset is employed to predict multiple time steps, by extrapolating the time series into the future [68]. To forecast, it is essential to ensure that the model can be trained with a sufficient number of observations relative to the parameters involved [58]. In this case, when a forecast of at least one year in advance is required, a historical training set of at least two years is imperative. Additionally, the historical data should encompass at least as many observations as the length of the longest anticipated seasonal pattern. Given the presence of an annual seasonality pattern, a minimum of three years of historical data is essential for this research.

A prediction model is designed to gain insights into the capacity needed in the future. The time series employed for the forecast models and the extreme value analysis, as described in Section 4.1, is the Internet Peering data shown in Figure 5.1. The data features the highest daily throughput recorded in the network between March 9, 2020 and October 1, 2023.

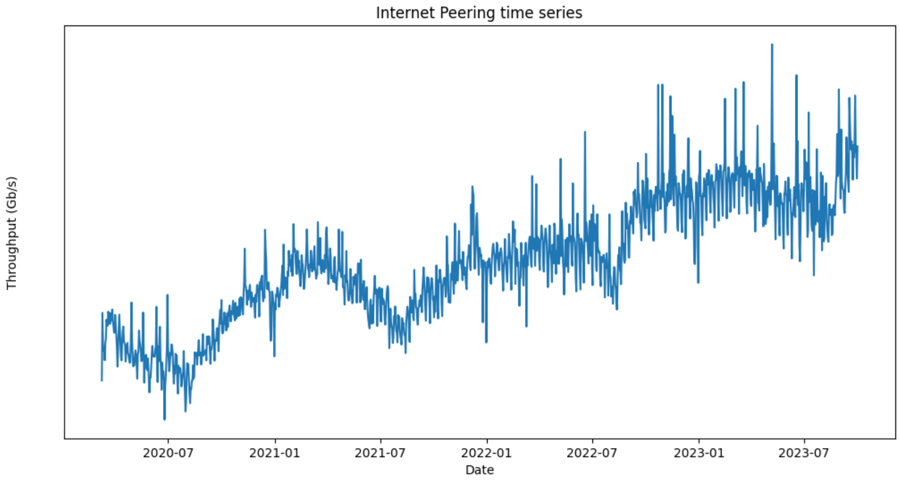
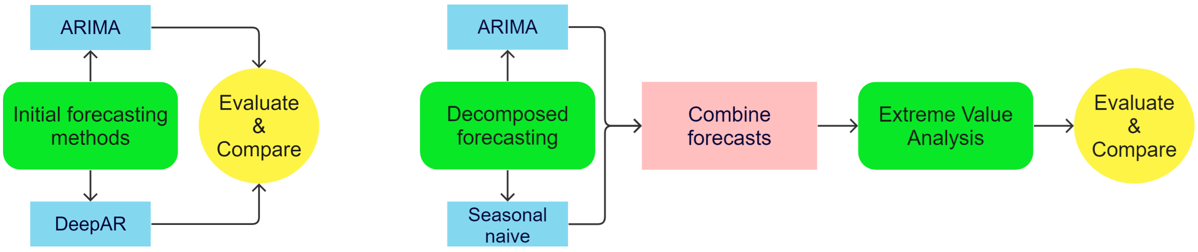


Figure 5.1: The daily maximum throughput of the Internet Peering component during a period of 3,5 years.

Figure 5.2 illustrates the evolution of the model development process, which unfolded in two distinct



(a) The initial forecast methods.

(b) The final chosen forecast methods used for the prediction model.

Figure 5.2: The steps taken in the development process towards a prediction framework.

phases. The initial concept, as depicted in Figure 5.2a was to develop a (machine learning) model with the ability to predict the maximum expected traffic with additional knowledge of extreme events. This external data encompassed anticipated events such as F1 races, World Cup and European Championship matches, and more. Two distinct methods were explored to evaluate the feasibility of this concept, as elaborated upon in Section 5.1.1. However, this concept proved infeasible as it did not yield accurate results for the extreme values within the time series. Consequently, an alternative approach was pursued.

Subsequently, in the second phase illustrated in Figure 5.2b, a decomposed time series forecast approach was implemented. Then this forecast is integrated with an EVA model, to obtain the forecasts on the extreme values of the dataset. This method leverages the statistical characteristics of the time series, to ensure reliable and interpretable forecasts that are essential given their role in network design decisions.

The results obtained from the forecast methods have been evaluated and compared to determine the subsequent steps necessary for the final design of the prediction model. MAPE has been employed to compare the performance of the prediction models [21]. This is a widely employed metric to assess the accuracy of the prediction method. The model that exhibits the lowest error value represents the optimal model, which indicates its ability to generate predictions that closely align with actual test data values. Equation 5.1 depicts the formula employed to calculate the MAPE:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{Y_t} \times 100 \tag{5.1}$$

Here t represents the specific period $(1, 2, 3, ..., n)$, n the total number of observations and Y_t the observed throughput at time t. The MAPE is calculated by taking the absolute difference between the predicted $\hat{Y}_t$ and observed values, relative to the observed values.

The various models have been assessed on complexity and reliability, to determine which methods are most applicable for real-time data. Moreover, this research aims to establish a framework for capacity management. Therefore the integration of DataIku has been considered in the design process. Further details regarding the usage of this platform are provided in the following section.

5.1.1. DataIku

Two prediction methods employed to evaluate the feasibility of the time series were examined and produced with the use of the DataIku platform. AutoARIMA and DeepAR have been applied in DataIku to predict the maximum daily throughput and their performance was assessed. These methods were chosen as they both allow the use of exogenous variables to improve the forecasting accuracy [69].
To design a prediction model, the complete time series spanning 1302 days, was divided into a training set and a test set, aligned with the intended forecast horizon of 365 days. This partitioning resulted in a training set of 937 days and a test set encompassing the most recent 365 days. It was not feasible to achieve a forecast horizon of six quarters due to the length of the dataset being 1302 days, falling short of the required 1644 days. This discrepancy arises from the training requirements of the model,

where a forecast horizon of 548 days necessitates twice that duration for training and an additional 548 samples for testing.

AutoARIMA

Automatic ARIMA (AutoARIMA) is a time series prediction approach that identifies the optimal parameters for the (S)ARIMA model [70]. It conducts a systematic search, with no constraints or user predefined constraints, to select the parameter set that minimizes the specified information criterion, such as the MAPE. This parameter set consists of the AR and MA model orders, the seasonal order and the season length, considering factors such as the unit root tests for stationarity detection. This will result in the highest possible accurate SARIMA model using the given time series for forecasting.

Two distinct tests were conducted to identify suitable parameters for the SARIMA model. In the first test, no constraints were placed on the minimum or maximum values of the parameters. AutoARIMA conducts a unit root test to determine the order of differencing d, which leads to the selection of the optimal p and q parameters. Only the season length m needs to be determined in advance. In order to incorporate the annual seasonality, it was initially desired to set a season length of $m = 365$ days. However, due to computational constraints arising from the frequency of the daily input data, this season length was not feasible. Consequently, the season length was adjusted to $m = 7$ days. This allows the model to take into account the weekly seasonality.
The second test involved iteratively selecting SARIMA parameters based on multiple tests that have been performed on the data, which have been performed in Python. The results of these tests helped to determine the model orders regarding the stationarity and autocorrelation lags. It also showed a weekly seasonality and therefore a season length of $m = 7$ days was validated. As explained in Section 4.2.3, the Internet Peering time series is non-stationary, which led to setting the differencing term d to 1.

The forecast horizon is set to 365 days or approximately one year. The SARIMA model from the autoARIMA approach without parameter constraints and the SARIMA model with determined parameters, resulted in the same SARIMA model. The parameter set led to the ARIMA(1,1,1)(2,0,2)[7] model. The forecast of the first model in DataIku is depicted in Appendix B.1. The results of the second forecast model are depicted in Figure 5.3. The model resulted in a MAPE of 16.5%.

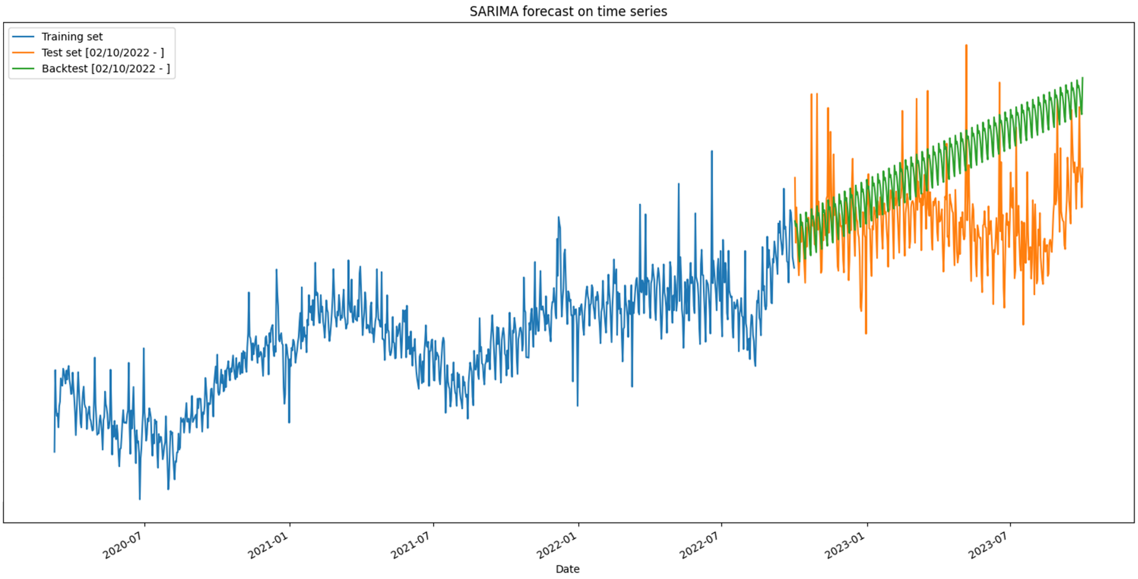


Figure 5.3: ARIMA(1,1,1)(2,0,2)[7] model on the Internet peering time series.

The backtest shows the performance of the prediction model when compared to the test set of the historical data. As weekly seasonality is considered, the forecast consists of the weekly traffic pattern. As a result, the trend and seasonality appear to be inconsistent with the expected growth based on the historical data. The results show that the model of ARIMA(1,1,1)(2,0,2)[7] does not provide an accurate forecast and therefore SARIMA is not a suitable approach for this time series input.

DeepAR

In addition, the time series data was used as an input for DeepAR. DeepAR is a time series forecasting model, developed by Amazon [71], designed to deliver accurate probabilistic forecasts for a wide range of applications. The model architecture consists of a stack of RNNs parameterized by specific features [44]. These are features that contribute to the learning process and they can be categorized as item-dependent, time-dependent, or a combination of both. The features allow the model to capture intricate relationships in the data. The model supports the use of exogenous variables, like Formula 1 race days, as additional input for training the model. Moreover, the approach goes beyond merely forecasting the future values of the time series. DeepAR also calculates the quantile estimates to provide a basis for more informed decision-making. It can learn from multiple time series simultaneously, which makes it suitable for datasets with numerous related sequences.

One model is trained using only the historical time series data. Moreover, another model is developed that incorporates exogenous variables. In this case, the exogenous variables use the Formula 1 race days as an extra input to train the model that peaks are expected on race days. This input consists of an additional binary input column, where a ”0” indicates no Formula 1 race has taken place and a ”1” indicates the occurrence of a Formula 1 race on that day. To obtain a forecast using these exogenous variables, future values of this feature must be given to the trained model to predict the throughput. Therefore, another column is added with the race days that will take place in the future, starting from October 2, 2023 until the end of 2024. The results of the DeepAR model trained with the exogenous variables are shown in Figure 5.4. Since this model incorporates exogenous variables, the figure displays predictions based on future exogenous input. A backtest was conducted, that yielded a MAPE of 16.3%.

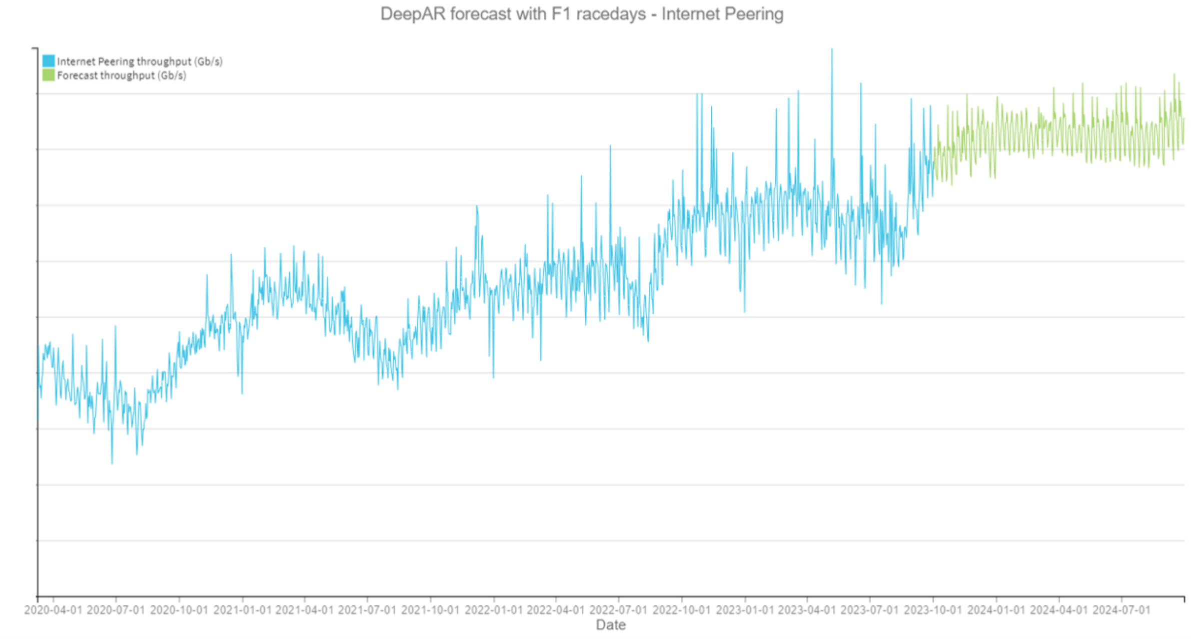


Figure 5.4: The results of the DeepAR model trained with F1 racedays [10].

The forecast of the DeepAR model without exogenous variables is elaborated on in Appendix B.2. The results show that the model trained with exogenous variables recognizes a higher trend and peak behavior on Formula 1 race days. The overall throughput is higher and the model shows an increased throughput on the race days, as shown in the appendix in Figure B.4. However, the MAPE of the model is still 16.3%. Although the overall predicted throughput is higher, the trend does not grow as expected as the magnitude appears to remain constant. Furthermore, the seasonality has disappeared. The absence of the seasonal component could indicate that information linked to this seasonality, such as summer and winter months should be used as exogenous input for the model. Also, the values on the race days do show an increased throughput but are not as high compared to the average throughput, as apparent for the high and sharp peaks in the historical data. It struggles to predict a significantly increased magnitude for the maximum throughput associated with race days. This challenge seems to be due to the limited data available for training, given that the race days account for only 6%, or 78 days out of 1302 days, and are divided between the training and test data sets.

DeepAR has a better capability in extracting intricate patterns and the recognition of extreme peak values from the time series, even when historical data is limited. This explains the improved forecast results that the model gives compared to the results of the ARIMA model. However, the time series forecasts conducted did not yield accurate predictions. The results of the DeepAR model imply that the model does not recognize the seasonality and fails to show extreme traffic peaks. As a result, the decision has been made to employ the decomposed time series for subsequent forecasting efforts. Additionally, an extreme value analysis will be incorporated to enhance the predictive capabilities for the extreme values.

5.2. Chosen forecast method with Extreme Value Analysis

To predict the throughput values of the Internet Peering time series, subsequent steps will be followed as depicted in Figure 5.5. The initial stage involves the extraction of the dataset from reports that contain the measured daily maximum throughput values. Subsequently, the dataset undergoes preprocessing to derive a time series suitable for the framework.

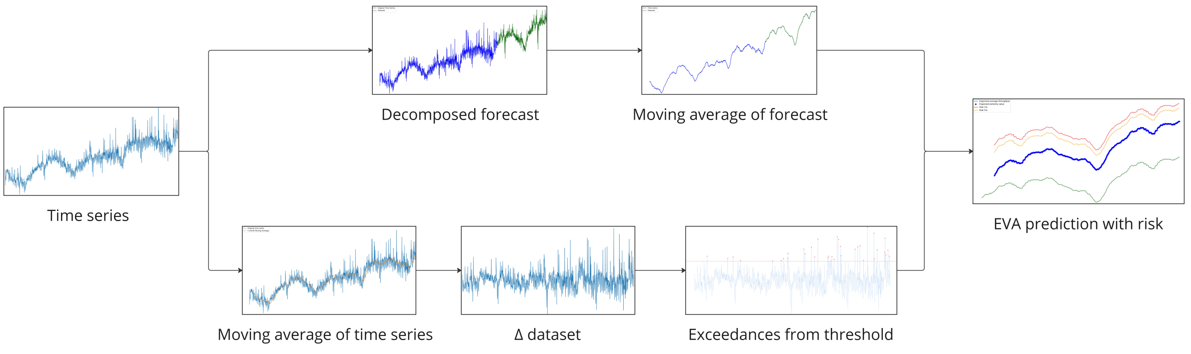

Time series
Decomposed forecast
Moving average of forecast
EVA prediction with risk
Moving average of time series
Δ dataset
Exceedances from threshold

Figure 5.5: The prediction framework with the chosen method of EVA.

Concurrently, two processes unfold. The first process is the calculation of the decomposed forecast. This includes the predictions for both the trend and seasonality components. Hereafter, a 31-day moving average is applied to the combined decomposed forecast. This moving average is essential to compute the absolute peak values with the results of the EVA model.
The second phase commences with the computation of the 31-day moving average for the original time series. This moving average is then utilized to generate the Δ dataset that encompasses the differences between the average throughput over the past month and the daily maximum measured peaks. This dataset is then used for the EVA model, for which threshold selection must be performed. A threshold is determined based on the distribution of this dataset and exceedances beyond this threshold are subjected to fit an EVA model. The model yields results for return levels, that represent the expected deviations from the average throughput.
Then the two processes converge. The 31-day moving average derived from the decomposed forecast is combined with the results of the EVA model. Consequently, the return levels are added to the moving average, which yields the absolute expected forecast values. For these extreme value forecasts, the model will also compute the risk levels at 5% and 1%. As a result, a prediction for 1,5 years in advance can be calculated.

5.2.1. Decomposed time series approach

Time series that contain extreme values often pose challenges for accurate forecasts. As discussed in the previous section, the employment of SARIMA or DeepAR models on the daily peak network throughput is likely to yield inadequate forecasts. Consequently, an alternative prediction model has been devised, that leverages the trend and seasonality. The multiplicative time series composition detailed in Section 4.2.3 unveiled distant characteristics, which allowed for the extraction of trend and seasonality components. These characteristics can also serve as valuable inputs to forecast the time series.The equation employed to perform a decomposed forecast, denoted as $\hat{y}_t$, is [65]:

$$\hat{y}_t = \hat{S}_t \cdot \hat{T}_t \tag{5.2}$$

$\hat{S}_t$ represents the forecast of the seasonal component of the time series. This has been predicted using a seasonal naive method, as the seasonality is consistent throughout the years. The seasonal naive forecast is calculated by Equation 5.3.

$$\hat{S}_{t+h} = S_{t+h-m} \tag{5.3}$$

Here $\hat{S}_{t+h}$ represents the forecast for period $t+h$, where h represents the number of periods to forecast. $\hat{T}_t$ is the trend component, which can be predicted using a non-seasonal method such as ARIMA. More on the decomposed forecast is described in Appendix B.3.

5.2.2. Extreme Value Analysis

EVA has been employed to enhance the decomposed time series forecast and to create a statistical foundation for the evaluation of the results of the machine learning forecasts. This can be achieved by the prediction of future extreme values and the period of occurrence of these throughput levels. Accurate prediction of these extreme values holds significant importance. They define the maximum capacity requirements of the network and thus determine the necessary design investments to meet these requirements. Consequently, a lack of accurate foresight into future demand could lead to the ceasing of the network to offer data and its services.

The extreme events are modeled using an extreme value distribution. Two methods can be used to analyze the extreme values. The first approach is the Block Maxima approach, which is based on the Generalized Extreme Value (GEV) distribution. The second method is the Peaks over Threshold (POT) approach, which is based on the Generalized Pareto Distribution (GPD) [72]. They are suitable for modeling the maximum or minimum values of a time series sample. In this case, the upper extremes are important.

Peaks over Threshold

The chosen method is the POT approach. Using the Block Maxima method may prove inefficient when one block contains a multiple number of extreme events. This results in the loss of useful data, as is demonstrated by the application of the Block Maxima method on the Internet peering time series in Appendix C. This loss can lead to less accurate estimates of extreme values thus potentially underestimating the true risk associated with rare events. Additionally, in case a block lacks extreme events, the Block Maxima method will still incorrectly label the highest value within that block as an extreme event. Consequently, this leads to an inaccurate representation of the true extreme values in the dataset. When working with a low granularity time series dataset, such as daily observations, a more efficient approach is to avoid the usage of block maxima and use the POT method [73]. Moreover, the POT approach is better suited for assessing tail losses, as it concentrates on the distribution of exceedances beyond a chosen threshold. This allows for a more detailed examination of extreme events.

The extreme events represent the realizations x of a random variable X, exceeding a sufficiently high threshold u. When X is characterized by the cumulative distribution function $F(x)$, the conditional excess distribution function $F_u(x)$ is stated in Equation 5.4, for exceedances X over a threshold [74] [75].

$$F_u(x) = P(X - u \leq x | X > u) = \frac{F(x+u) - F(u)}{1 - F(u)} \tag{5.4}$$

As the threshold gets large, the Gnedenko-Pickands-Balkema-deHaan (GPBdH) theorem states that the distribution converges toward a GPD [76]. The cumulative distribution function of the GPD is defined in Equation 5.5 [77].

$$F(x) = 1 - \left(1 + \frac{\xi(x-u)}{\sigma}\right)^{-1/\xi} \quad \text{for } x \geq u, \xi \neq 0 \tag{5.5}$$

The GPD has three parameters; the threshold u. the scale parameter σ and a shape parameter ξ, which controls the tail weight of the distribution. The threshold u should be chosen high enough for the

GPBDH theorem to be applied and should include a sufficient number of observations that exceed the threshold value. It is essential to choose a value that is not too high as this results in an insufficient number of extremes, which gives unreliable estimates. Selection of thresholds can be done in various manners; graphical and numerical approaches [72]. More on how this threshold is selected for this model is described in Section 5.2.2.
POT requires mutually independent extreme values. Therefore declustering can be used to filter dependent observations so a set of extreme values can be obtained that are independent. The fixed threshold will determine the extreme values and then a minimum length is set between each cluster, to define every separate cluster. In this case, this length is set to one day as the daily data needs to be analyzed for every possible peak value. Every extreme will be identified according to the set threshold and then the GPD will be fit to the independent maxima.

To build a time series model with Extreme Value Theory, the next steps are taken:

1. Dataset of relative differences: A new dataset was created to facilitate the POT method for the time series.
2. Threshold selection: A threshold is chosen that defines the data points considered as extreme events. For this threshold the POT method is used, to ensure minimum data loss. The extreme events are identified and extracted from the time series data for further analysis.
3. Model fitting and validation: As the POT threshold approach is chosen, the GPD is used to model the extreme values. The GPD has been fitted to the extracted exceedances from the time series. This involved the parameter estimation of the GPD that best fit the data with the maximum likelihood (MLE) method [78]. Then diagnostics plots are employed to evaluate the fitted model.
4. Monte Carlo Simulation for uncertainty estimation: A Monte Carlo simulation is conducted by resampling the data to generate different datasets. For each dataset, the GPD is refitted to obtain a new set of parameter estimates. This assesses the variability of the parameter estimates and calculates a confidence interval for the return levels.
5. Return level estimation: The fitted distribution is used to estimate the probability beyond the observed data range. This involves the calculation of the return levels corresponding to the specific return periods. Moreover, risk level estimates are computed with the return periods associated with these risks.

More on this is elaborated on in the next subsections.

Threshold selection

To determine the threshold, four methods were selected for the relatively short time series that spans three and a half years. The first method was chosen from various graphical threshold techniques. Graphical threshold techniques have resulted in uncertainty and subjectivity [79]. Therefore, solely the Mean Excess (ME) plot was employed to help in the selection of the threshold. The plot aids in the selection of an appropriate threshold and the evaluation of the adequacy of the GPD model for the generated dataset. The ME plot is a visual representation that depicts the empirical ME function of a random variable X following a distribution F [80]. It is defined as stated in Equation 5.6, where $M(u)$ is the ME function at threshold u [81]. It represents the expected excess of X over threshold u, given that X exceeds u.

$$M(u) = \mathbb{E}[X - u \,|\, X > u] \tag{5.6}$$

In the case of an independent and identically distributed (iid) sample $(X_1, X_2, \ldots, X_n)$, an inherent estimate based on the actual data points for this function is the empirical ME function $\hat{M}(u)$ shown in Equation 5.7.

$$\hat{M}(u) = \frac{1}{n} \frac{\sum_{i=1}^{n}(X_i - u) \cdot I(X_i > u)}{\sum_{i=1}^{n} I(X_i > u)}, u \geq 0 \tag{5.7}$$

Figure 5.6 shows the ME plot for the Internet Peering time series. The plot represents the thresholds at which the ME function is evaluated. As the threshold increases, there is a steep decline in the ME values which then stabilizes. This range is where the ME values do not change significantly and can be considered for the threshold selection. Beyond this range, there is more variability in the ME values and the plot declines again. This suggests that there are few extreme values above these high thresholds, which would lead to an unreliable estimate of the extreme value distribution. The shape from the top left to the bottom right represents a decreasing trend that approaches zero for higher thresholds. This implies that the shape parameter of the GPD is negative, $\xi \leq 0$ and the distribution has a bounded tail.

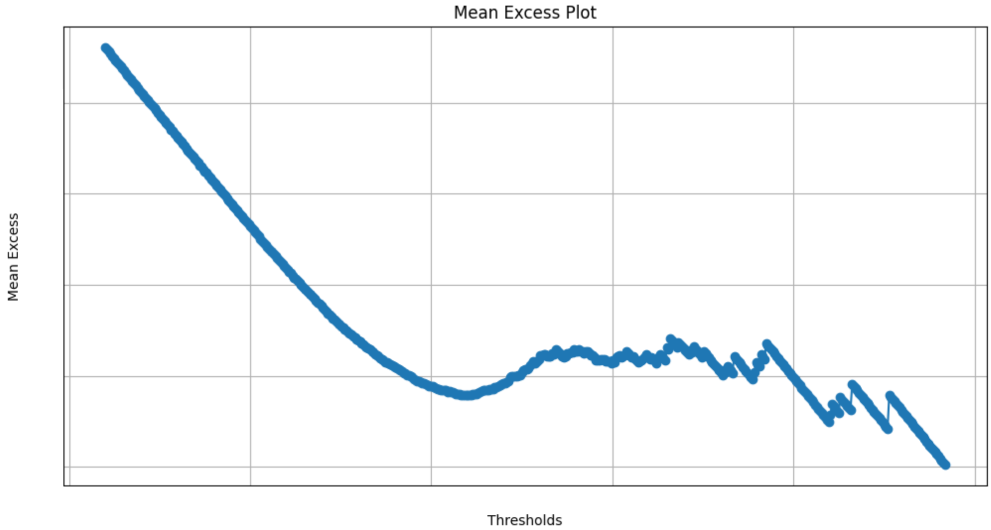


Figure 5.6: The ME plot of the Internet Peering time series.

The other three methods have been chosen for their simplicity and effectiveness in various contexts. The first one employs the upper whisker of the boxplot to identify outliers, based on the interquartile range (IQR), which measures the middle 50% spread of the data [82]. It is calculated as the difference between the third quartile and the first quartile. The upper bound is calculated as:

$$u_b = Q3 + 1.5 * \text{IQR}$$

This method is robust, non-parametric and based on a statistical measure that is less sensitive to extreme values. It is a common rule and it effectively captures points that are outside the normal range of the bulk of the data [82].
The second is the upper 10% quantile, which involves ordering the data of the time series in ascending order and calculating the value at the 90th percentile:

$$u_{10\%} = Q_{0.9}$$

This quantile-based approach guarantees that 10% of the most extreme data points will be considered as potential extremes, regardless of the distribution of the data. This also helps to ensure a sufficient number of extreme values for the POT model due to the fixed amount of exceedances.
Lastly, the threshold is chosen based on the square root of the sample size of the data $k = \sqrt{n}$ [83]. The top largest k values are ordered and the kth lowest value from these values is chosen as the threshold:

$$u_k = \text{min}(\{x_{(1)}, x_{(2)}, \ldots, x_{(k)}\})$$

This method should enable the estimation of a stable threshold, which is sufficiently high to exclude non-extreme values whilst capturing adequate extreme values for robust analysis.

Table 5.1 shows the thresholds calculated based on the three approaches.

Method	Threshold	Number of exceedances
Upper Whisker	u_b	36
Upper 10 % Quantile	$u_{10\%}$	128
$k = \sqrt{n}$	u_k	35

Table 5.1: Summary of thresholds.

The upper whisker quantile was selected as the final threshold for EVA due to its robust and standardized approach, which does not assume any specific underlying distribution. It includes a sufficient amount of data points for the EVA model and excludes non-extreme values of the distribution. Moreover, the threshold value indicated by the upper whisker aligns with the stabilization range in the ME plot, where the average size of excesses over the threshold captures the tail behavior of the distribution. This method provides a non-parametric, common statistical approach to identify extreme values. This threshold was employed to extract the extreme values from the dataset, that surpass this threshold value. Then the GPD has been applied to estimate the tail behaviour of the distribution.

Return level estimation

The objective of the EVA model is to present the maximum peaks, or return levels, that are expected to occur in the future. The return levels of the distribution represent the magnitudes of the extreme values, that can be reached within a specific return period. The return period T denotes the average amount of time it takes for a specific event to occur again [73]. This is crucial to determine the required network capacity.

To compute the return level, it is necessary to determine the probability of observing an extreme event in the future. The probability distribution of an extreme event given that variable X has exceeded threshold u is expressed in Equation 5.8, for $x > u$ [73].

$$\Pr\{X > x | X > u\} = \left[1 + \xi\left(\frac{x-u}{\sigma}\right)\right]^{-1/\xi} \tag{5.8}$$

The return level is the level anticipated to be reached approximately once every N years. If there are n_y observations annually, this corresponds to the N-year return level defined as follows [73]:

$$z_N = u + \frac{\sigma}{\xi}\left[(Nn_y \cdot \zeta_u)^{\xi} - 1\right], \xi \neq 0 \tag{5.9}$$

In Equation 5.9, $\zeta_u = \Pr\{X > u\}$ represents the exceedance probability of an individual observation over threshold u. To estimate the expected return level, the GPD parameters need to be replaced by their respective maximum likelihood estimates. MLE is a statistical method employed to estimate the parameters of a given probability distribution based on observed data. This technique involves the identification of parameter values that maximize the likelihood function, to ensure the most probable parameter set that results in the observed data for the chosen distribution. The Likelihood function, denoted as $L(\xi, \sigma)$, is the product of the probability density functions for each observed data point. The function is given by [73]:

$$L(\xi, \sigma) = \prod_{i=1}^{n} f(x_i | \xi, \sigma)$$

The Log-Likelihood function of the GPD is then described as in Equation 5.10.

$$\ell(\xi, \sigma) = \log L(\xi, \sigma) = -n \ln \sigma - \left(1 + \frac{1}{\xi}\right) \sum_{i=1}^{n} \ln\left(1 + \xi\frac{x_i}{\sigma}\right) \tag{5.10}$$

The ML estimates are then determined by differentiation and solved numerically in Python. The estimator of ζ_u is defined as follows:

$$\hat{\zeta}_u = \frac{k}{n} \tag{5.11}$$

where n denotes the total number of observations and k denotes the number of observations that exceed the threshold. As the observed data points are independent and identically distributed (i.i.d.), the number of exceedances above the threshold follows a binomial distribution Bin(n, ζ_u) and the estimator in Equation 5.11 therefore serves as the maximum likelihood estimate for $\zeta_u = \Pr\{X > u\}$.

Monte Carlo simulation is a technique that enables the estimation of real-world scenarios where outcomes may deviate from expectations. The method can be employed in EVA to explore the variability of model predictions. It serves as a tool to comprehend and quantify the variability in return levels

through confidence intervals. The confidence intervals characterize the range of maximum values observed within a period.
Each iteration of the Monte Carlo simulation involves the generation of a dataset from the observed extreme values and re-estimation of the model parameters with MLE. For each dataset, return levels are calculated based on the probability distribution defined by the model for the specified return periods. This calculation leverages the CDF of the fitted extreme value model. Each iteration yields a potential return level, based on the newly estimated parameters derived from each sampled dataset.
Consequently, this process results in a distribution of return levels from which a 95% confidence interval is derived, to quantify the uncertainty associated with the estimated return levels. The confidence interval is computed by the identification of the 2.5th and the 97.5th percentiles from the return level distribution [84]. The 2.5th percentile is the value below which 2.5% of the simulated return levels fall and the 97.5% percentile represents the value below which 97.5% of the values are observed. Collectively, these percentiles encapsulate the central 95% of all values within the distribution. This range therefore defines the confidence interval, which indicates a 95% probability that the true maximum return level will reside within this range. This offers a measure of the predictive reliability of the model and the uncertainty in extreme value predictions.

Risk analysis

For capacity planning it is important to balance the amount of capacity to install, and the costs associated, with the probability of a capacity shortage and potential network outages, thus the risk associated. Therefore it is insightful to understand the maximum value that can be surpassed with a specified level of risk. To determine this, the computation of return levels associated with a designated risk percentage is required. For this, the inverse of the return period $T = 1/p$ in years is employed, where p is the annual exceedance probability [85].

Suppose the risk of a capacity shortage over one year has to be limited to 1%. To determine the return level with a shortage risk of 1% in one year, the $1/0.01 = 100$ return value z_{100} (which has a return period of $T_{risk} = 100$) has to be calculated. Hence, to limit the risk of a capacity shortage over the next year to 1%, it is estimated that a capacity with value z_{100} has to be available in the network.
In general, if the risk of capacity shortage has to be limited to probability p% over a specific planning horizon H, the T_{risk}-year return level $z_{T risk}$ can be calculated with Equation 5.12.

$$T_{\text{risk}} = \frac{H}{p} \tag{5.12}$$

The chosen return period size for the horizon of the model is one year, which conforms to the average duration in the Gregorian calendar (365.2425 days). Therefore, a return period of 50 corresponds to a 50 year event and a return period of 0.5 represents a half year event. Similarly to the 1% risk level calculation, for the return level with a 5% risk in one year, the calculation is $1/0.05 = 20$ which leads to a return period of $T = 20$ a required capacity of z_{20}.
The return levels that result from the risk calculation will be linearly interpolated to establish the risk for each day of the quarter. Hereafter, the risk levels for each maximum expected throughput per quarter can be derived.

6 Results

This chapter outlines the framework of the prediction model that was designed with the Internet Peering data in the development process. This model for extreme values is tested on the Internet Peering time series and its results will be described. Moreover, these predictions are employed to challenge the manual forecasts. Furthermore, the framework is applied to the Mobile Core time series.

6.1. Framework of the model

The design process has resulted in the development of a structured framework that serves as a guide to generate predictions. Figure 6.1 visually encapsulates all the steps that comprise the final framework that is employed to forecast the time series data, as elaborated upon in the methodology chapter:

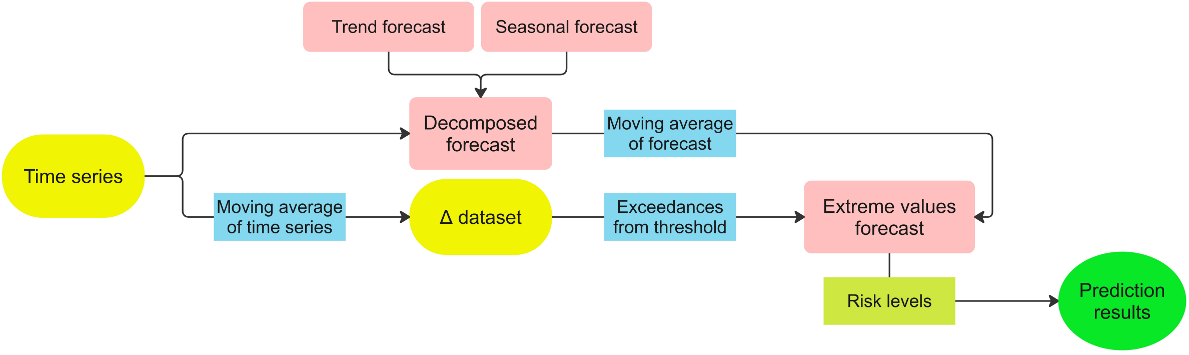


Figure 6.1: The prediction framework to forecast the maximum expected throughput values.

6.2. Decomposed forecast

The decomposed forecast is built on the forecasts of the separate components from the time series; the trend and seasonality. The forecast results of the trend and the seasonality can be seen in Appendix B.3.1. These predictions are multiplied to obtain the naive time series prediction for the average throughput calculation. The naive time series forecast results can be seen in Figure 6.2. The figure shows the multiplied trend and seasonality of the time series, with the combined decomposed forecast for six quarters in advance.

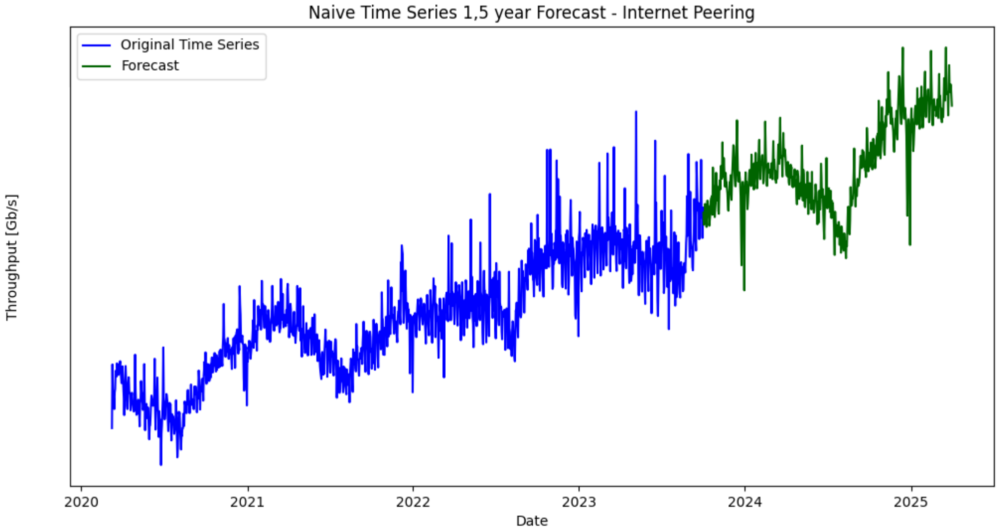


Figure 6.2: The naive time series forecast with a forecast horizon of six quarters of the internet peering traffic.

The models of the decomposed forecast show that an accurate prediction for the average throughput can be calculated with a MAPE of 2.0% for the ARIMA model of the trend and a MAPE of 0.0% for the seasonality prediction. The moving average of 31 days of the prediction is then calculated, which results in the prediction shown in Figure 6.3.

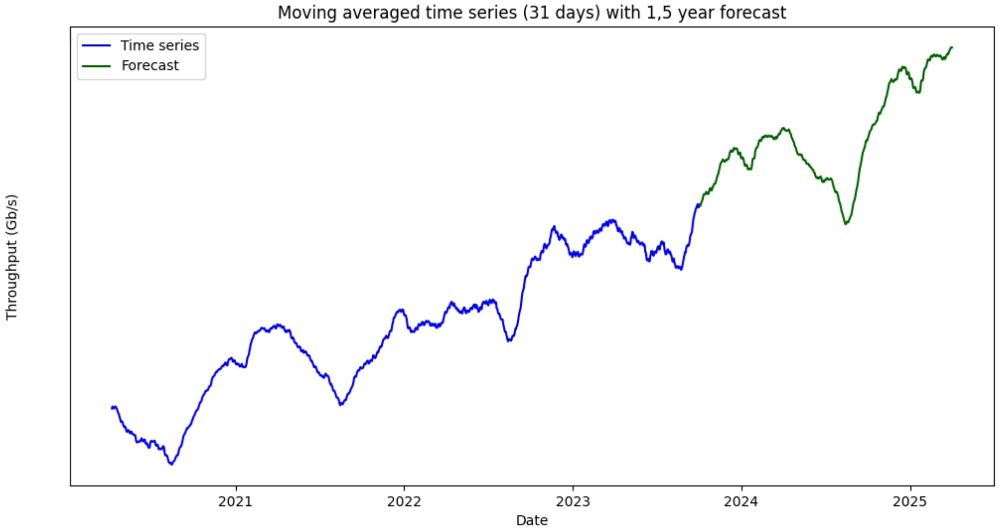


Figure 6.3: The moving averaged forecast of the internet peering time series.

The observed pattern of the prediction is consistent with the throughput trends of the time series that were previously outlined.

6.3. Extreme Value Analysis

The input data that is used for the EVA is the computed dataset of Δ_i depicted in Figure 4.3, which contains the deviations from the average throughput to the daily peak. The distribution of the dataset is visualized in Figure 6.4. The plotted graph of deviations against their corresponding frequency provides a detailed analysis of the deviations of the dataset. It enables the identification of the frequency of occurrence of calculated deviations, which helps to discern the tail of the dataset. The plotted graph reveals that the distribution of the dataset is reminiscent of a normal distribution, which is characterized by a bell-shaped curve.

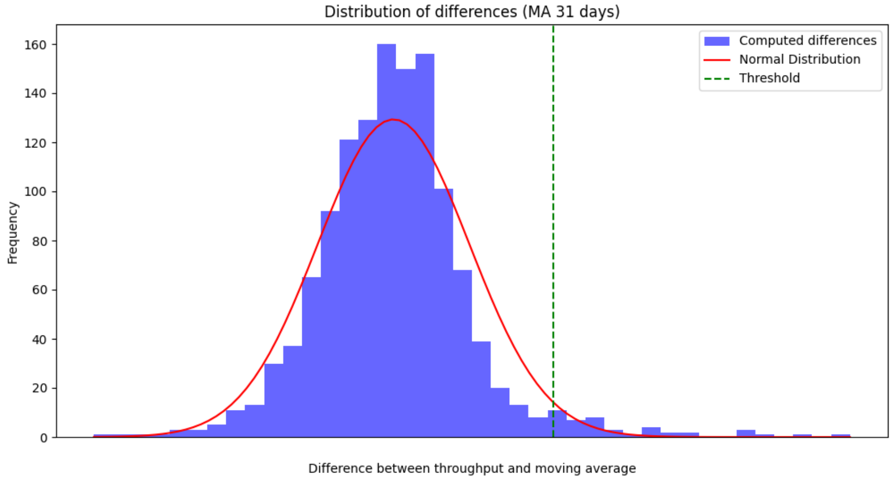


Figure 6.4: The distribution of differences and the threshold for the Internet Peering time series.

The threshold for the extreme values of this dataset was calculated at u_b in Section 5.2.2 and the green dotted line on the graph represents this value. The threshold determines which observations are considered extreme values and are used to estimate the GPD to model the right tail of the distribution of the observations.

Figure 6.5 displays the deviations dataset with the calculated threshold. The graph showcases the extreme values that exceed this threshold, thus qualify as exceedances, and illustrates a temporal correlation in the increase of their magnitude. A total of 36 extreme values were identified for this threshold and will be employed for the EVA model.

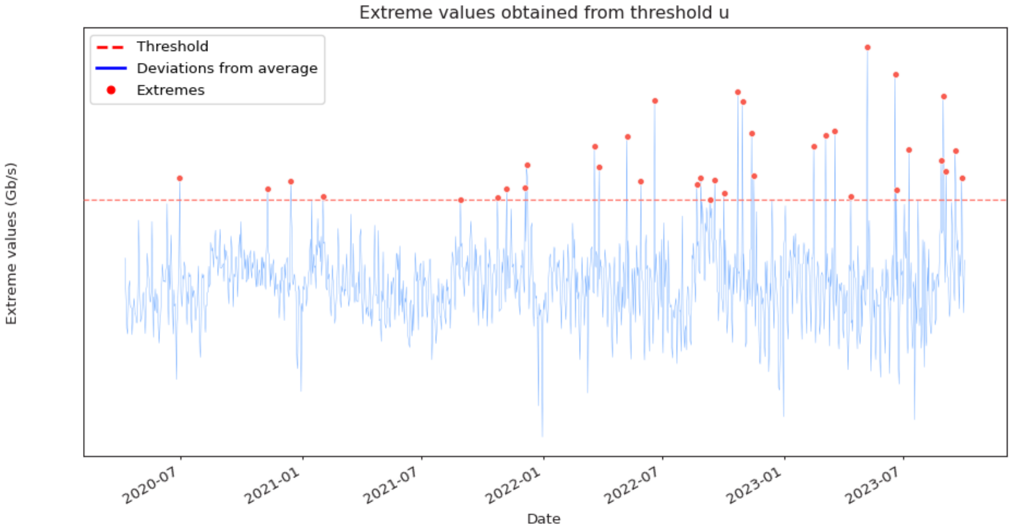


Figure 6.5: The selected extreme values of Internet Peering for threshold u_b.

Then the process involves the use of MLE to determine the GPD parameters from the observed data and the model is fitted to the extreme values. The main goal is to identify the parameter values that maximize the likelihood function and define its shape and scale. The threshold plays a crucial role in the determination of the GPD parameters, where the first parameter is the threshold u. The MLE computations have resulted in a scale parameter of $\sigma = 298.907$ and a shape parameter $\xi = -0.173$. To evaluate the designed EVA model with these estimated parameters, the diagnostic results of the fitted model are shown in Figure 6.6 and discussed below.

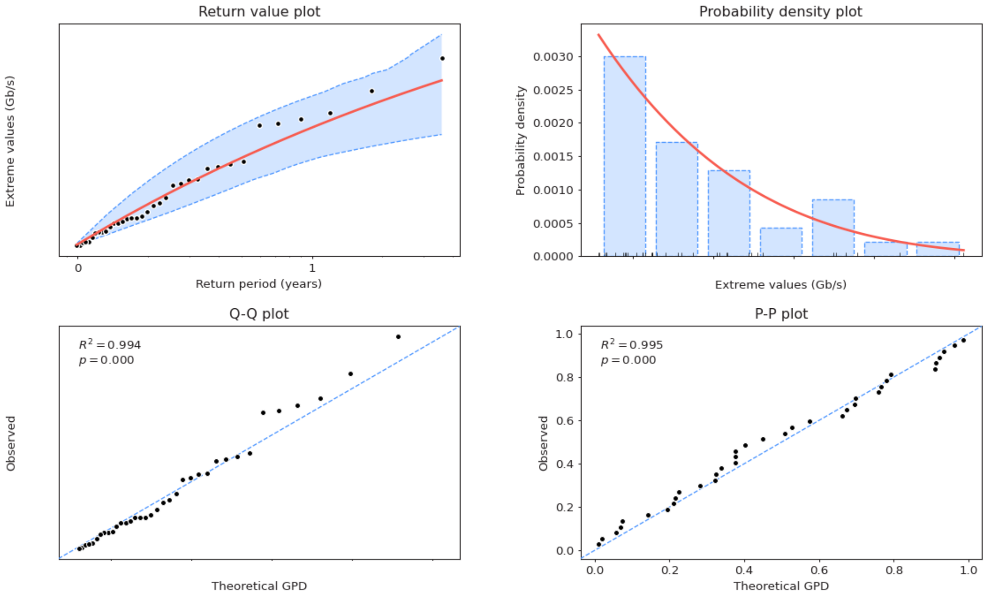


Figure 6.6: The diagnostics for the EVA model for Internet Peering.

- **The return value plot**: The return level plot shows the relationship between the 36 extreme values and their corresponding return periods. These are calculated based on the number of exceedances within the given period of observation. The extremes are ranked in ascending order from 1 to n. With the Python function *pyextremes* [86], the ranks are used to find the exceedance probability for each extreme value. The return period is then calculated from the exceedance probabilities and the rate of extreme events. This depends on the number of extremes, in this case 36 and the total duration of the time series from which the extremes were drawn. The red line represents the expected theoretical values from the GPD model and the black dots are the observed values of the dataset. The blue region depicts the 95% confidence interval for the variability of the return levels. The observed extreme data is resampled for a thousand runs by a Monte Carlo simulation, to compute confidence intervals for the return levels.
- **The probability density plot**: The probability density describes the likelihood of observing a particular extreme value. The extreme values that align along the line, represent the conformity to the GPD. Deviations from this line may indicate deviations from the GPD, however this is expected with an estimation of a distribution. It can be seen that around the extreme values with a higher magnitude, the probability density of the observed extreme values is higher than the fitted GPD probability distribution. This can indicate that the extremes with this magnitude are not as rare as expected for the GPD and therefore deviate from the fitted GPD.
- **The Q-Q plot**: The Q-Q plot is an empirical tool to compare the quantile of each observation and the predicted quantile. The plot facilitates the evaluation of the goodness of the fit of the parameterized GPD model to the empirical data [87]. R^2 is the correlation coefficient that quantifies the degree to which the observed data points align with the expected theoretical quantiles of a probability distribution of the GPD.

- **The P-P plot**: This plot relies on an assumed probability for the measured events. It is assumed that the largest observed event has the lowest probability of occurring. R^2 indicates how closely the empirical CDF of the data matches the CDF of the fitted GPD. The resulting R^2 coefficients indicate that the theoretical distribution provides a good approximation of the distribution of the extreme values.
  From the diagnostics can be concluded that the GPD model is a suitable fit for the extremes of the daily throughput traffic.

As the model is fitted to the data, the expected return levels for the next six quarters can be computed. The return periods for each day over the next six quarters have been determined and used to calculate the return levels that correspond to the return periods. The graph illustrated in Figure 6.7 shows the calculated return levels from the GPD model fitted to the extreme values. The EVA return levels of the blue line represent the predictions of the expected deviations from the average throughput for the next six quarters. The return levels do not define the absolute maximum expected traffic peaks, but only the deviations from the 31-day moving average.

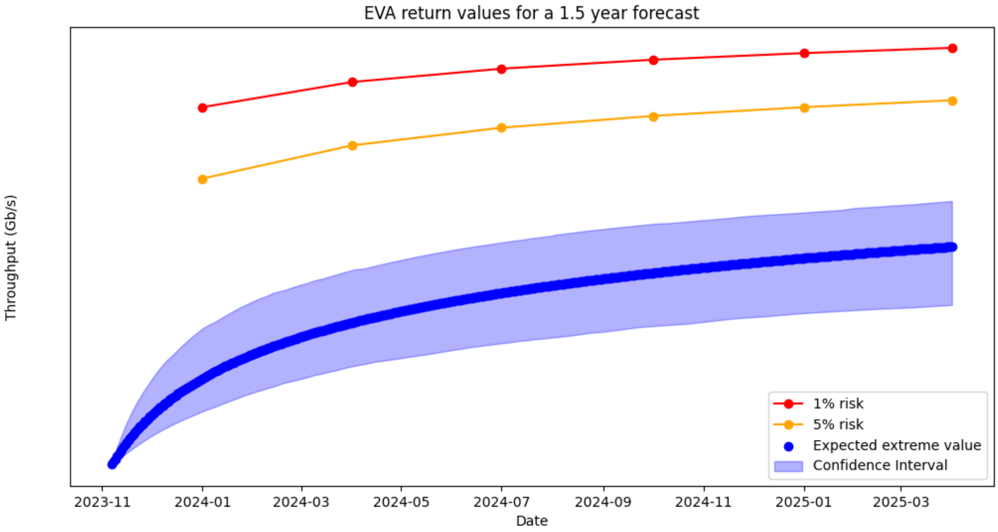


Figure 6.7: The unprocessed results of the EVA model for the next six quarters of Internet Peering.

As stated before, a Monte Carlo simulation was performed to estimate the variability of the return levels. Although the model assumes the data points are i.i.d., the observed exceedances indicate an increase in the magnitudes of peaks over time. To quantify this variability, $n = 1000$ bootstrap samples of the extreme observations have been generated. This creates 1000 datasets based on the 36 extreme observations in the original dataset.
This method recognizes that the frequency of extreme events within the next six quarters may vary from the expected 36 extremes, which can significantly influence the likelihood of specific return values. Variability arises from a random selection of occurrences from the statistical distribution. While the expected return values have already been computed with Equation 5.9, it is imperative to consider potential variations in the distribution of these return levels.

Moreover, the expected return levels z_{Trisk} corresponding to a 5% risk and 1% risk of occurrence have been calculated. For each quarter in the future, the return periods associated with the risk levels were employed to obtain these return levels. Table 6.1 yielded from the equations of the return periods. The specified return periods can be used to calculate the return values with the associated risk levels with Equation 5.9. These values are significantly higher than the expected return values, as can be expected.

Table 6.1: The return periods for the risk calculations.

Period of Occurrence	Risk (%)	Return Period (Years)
1 Quarter = 0.25	5 1	5 25
2 Quarters = 0.5	5 1	10 50
3 Quarters = 0.75	5 1	15 75
1 Year = 1	5 1	20 100
5 Quarters = 1.25	5 1	25 125
6 Quarters = 1.5	5 1	30 150

The predicted EVA return levels are then added to the forecast of the 31-day moving average, to obtain the absolute expected traffic peaks. Figure 6.8 represents the absolute results of the prediction framework, where the EVA results are combined with the moving average forecast. The resulting expected extreme values with confidence interval can be seen in Figure C.2 in the appendix.

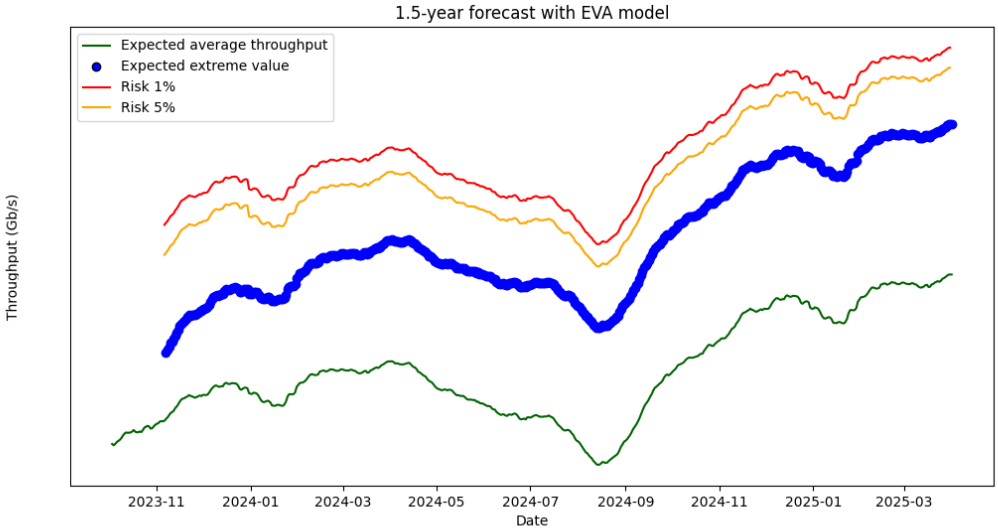


Figure 6.8: The results of the EVA model with risk levels for Internet Peering.

The graph shows the moving average forecast, along with the expected maximum throughput that is represented by the blue dots. To interpret the results, it is important to keep in mind that the calculations for the extreme values were performed with input data up until October 1st, 2023. From this point of view, the prediction results can be read. The predictions do not give the expected peak value to occur on a specific day. The blue dots describe the maximum peak expected to occur within the period until that specific day, if the value were to occur on that specific day as it is added to the expected average throughput of that specific day. This means that between October 1st and any specific day in the future, the maximum throughput calculated for that day in the graph can be expected to occur within the period until that day. The model expects the first values above the threshold to occur between October 1st and November 2023 and from this date on, extreme values are expected.

Moreover, the absolute risk values are depicted. The 5% and 1% risk values have the same interpretation as the expected maximum throughput but with different chances of occurrence. The values for the 5% risk have a 5% chance of being reached between October 1st and the corresponding date in the future. This means that there is a 95% chance that the throughput will not reach this value within that period. The same holds for the 1% risk extreme values. For these values, a chance of 99% can be assumed for the throughput not to reach the 1% line.
For example, if the capacity needs to be determined with the requirement of a 5% chance of a capacity shortage in the first semester of 2024, the maximum capacity described by the risk level of 5% within this semester should be considered. This would be the return value of the 5% risk level, which is the value around April 2024.

In the return values estimated by EVA, seasonality defines the maximum expected peak for Q2 and Q3. In Q2 the expected throughput decreases and in Q3 the expected throughput reaches its low. Then at the end of Q3, the throughput rises again in September with an upward trajectory in Q4. This indicates that the maximum expected throughput value between the required capacity in Q2 and Q3 does not change much.

The model calculates the maximum reached throughput per quarter based on the results. The same holds for the risk values, where the highest expected risk values are computed per quarter.
The results are compared to the manually calculated predictions. These predictions are the numbers that are presently used to determine whether equipment must be built for capacity. The forecast extends until Q4 2024 since there is no prediction for Q1 2025 yet. The difference $\Delta\hat{Y}$ between the current prediction and the EVA estimation is calculated in Table 6.2, in total throughput and percentage.

Table 6.2: The current manual forecast calculations compared to the results of the EVA model.

Quarter	Manual prediction (Gb/s)	EVA estimation (Gb/s)	$\Delta\hat{Y}$(Gb/s)	$\Delta\hat{Y}$ (%)
Q4 2023				-4,4%
Q1 2024				0,6%
Q2 2024				-2,9%
Q3 2024				-7,2%
Q4 2024				-2,7%

The results of the manual predictions increase per quarter. Except for the throughput values of Q1 2024, the manual calculations result in a higher throughput than the maximum expected throughput from the EVA. This could indicate that the manual calculations use information that estimates the throughput to increase more than the statistical characteristics provide. Moreover, no seasonality is seen in the manual predictions.

Table 6.3 depicts the same results where the EVA and risk values have been rounded up to represent the required throughput values for the estimation of extra capacity. This is because the service routers in the network are built with 100 Gigabit ports, which allows them to transmit up to 100 Gb/s. The EVA estimates show small differences compared to the manual predictions. The difference can be attributed to the seasonality factor, which was not considered in the manual predictions. In terms of the number of ports, this difference would mean that four ports in the service router would be required less per the EVA estimates. This deviation can not be considered small, as four ports are a significant investment. The EVA calculations suggest that this extra capacity is not expected and thus not needed, which makes it a more efficient calculation. This concludes that the total increases estimated by both manual and EVA predictions are comparable. The EVA predictions are based on substantiated calculations and demonstrate that it is possible to predict the throughput by analysis of the time series data.

Table 6.3: The current manual forecast calculations EVA estimates rounded up for implementation.

Quarter	Manual prediction (Gb/s)	EVA estimation (Gb/s)	$\Delta\hat{Y}$(Gb/s)	$\Delta\hat{Y}$ (%)
Q4 2023				-3,8%
Q1 2024				1,9%
Q2 2024				-1,8%
Q3 2024				-6,9%
Q4 2024				-1,6%

The risk predictions at the 5% and 1% levels offer additional insights for capacity planning, beyond the maximum expected throughput values from the manual predictions. The risk estimates reveal that the initial gap between the expected capacity based on manual predictions and the risk values is substantial. Similar to the expected peaks, the risk values from the EVA model consider the seasonality of the dataset and explain this small difference. In terms of capacity expansions, this would mean a difference of one port or three ports for the transport core links. The risk estimates suggest that the capacity expansions could be performed later which could improve the cost-efficiency trade-off.

6.4. Mobile Core implementation

The framework should be implemented for all different domains. To evaluate the designed framework of the Internet Peering time series for another domain, the time series data of Mobile Core has been employed as input for the prediction model. Therefore, all steps as described in Section 6.1 were implemented to calculate the predictions.

Firstly, the decomposition was performed for a window length of 365 days. The results of the decomposition are depicted in Figure 6.9. These time series components can then be employed for the decomposed forecast.

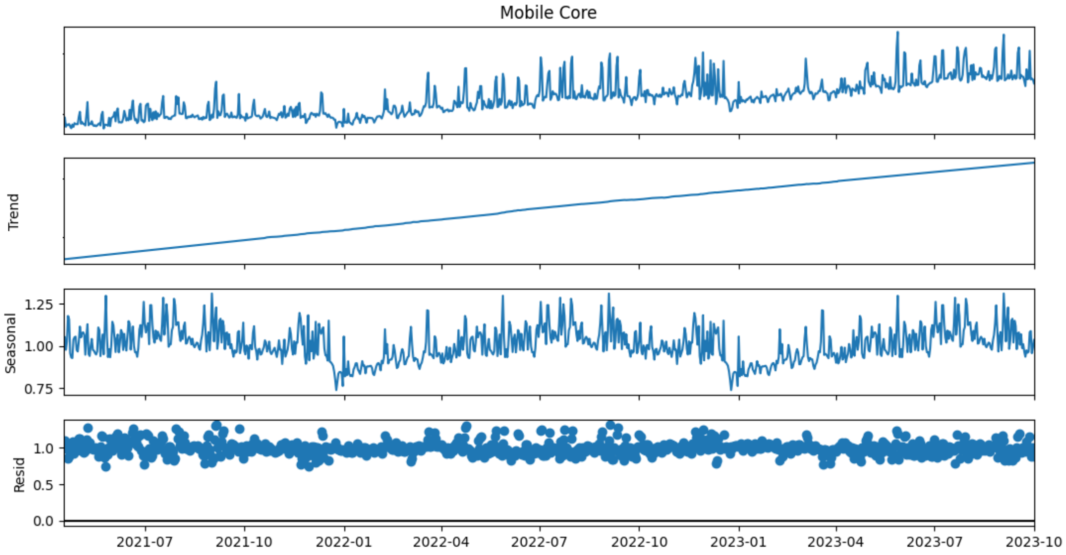


Figure 6.9: The multiplicative decomposition of the Mobile Core time series.

The time series of Mobile Core contains 896 days. However, to obtain a model with a forecast horizon of six quarters in advance, a time series with a minimal length of 1096 days is required. Therefore, until the dataset contains enough data points, a workaround has been created to compute the predictions. Instead of six quarters in advance, a prediction horizon of one year is used. When the time series has 1096 measurements, the original prediction framework can be employed again to estimate the throughput for six quarters in the future.

The decomposed forecasts for one year in advance are shown in Appendix B.3.2. The moving average of 31 days of the prediction has been calculated to obtain the prediction needed for the EVA model. These results are depicted in Figure 6.10.

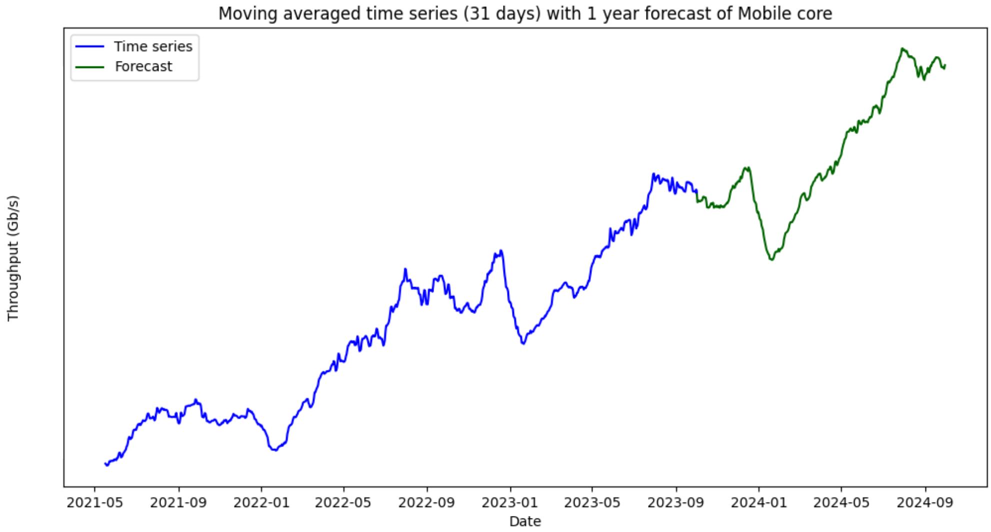


Figure 6.10: The moving averaged forecast of the Mobile Core time series.

Secondly, the dataset that contains the difference between the daily peak values and the average throughput was created. Figure 6.11 shows the dataset with the values of Δ_i.

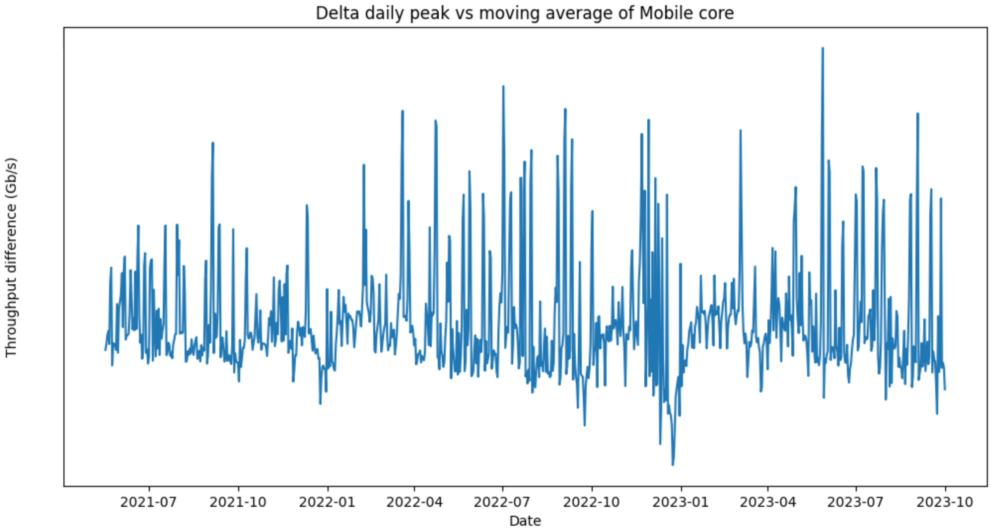


Figure 6.11: Generated dataset Δ_i of the Mobile Core throughput.

The calculated threshold yielded a value of u_b and resulted in the 48 extreme values as depicted in Figure 6.12.

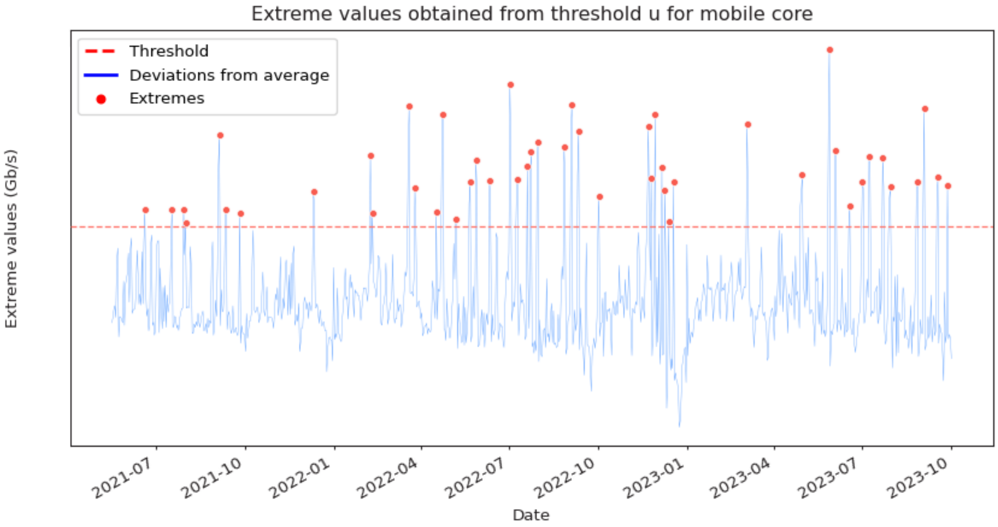


Figure 6.12: The selected extreme values of Internet Peering for threshold u_b.

The MLE computations produced a scale parameter of $\sigma = 52.084$ and a shape parameter $\xi = -0.426$. The diagnostics of the model are shown in Figure 6.13. The R^2 correlation coefficient shows that the GPD model is a suitable fit for the extremes but it is lower than the R^2 of the Internet Peering time series.

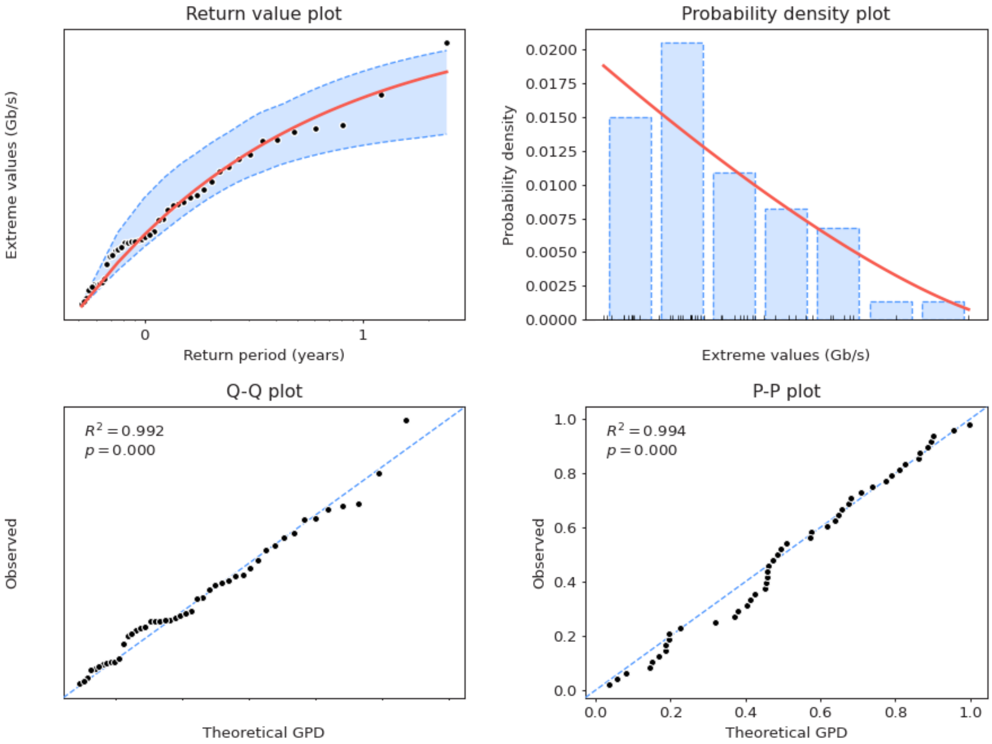


Figure 6.13: The diagnostics for the EVA model for Mobile Core.

The 48 extreme values are used to calculate the EVA return levels with the fitted GPD model. This results in the predictions for the next four quarters, illustrated in Figure 6.14. Moreover, the risk calculations have been performed for the return levels associated with a 5% and 1% risk of occurrence.

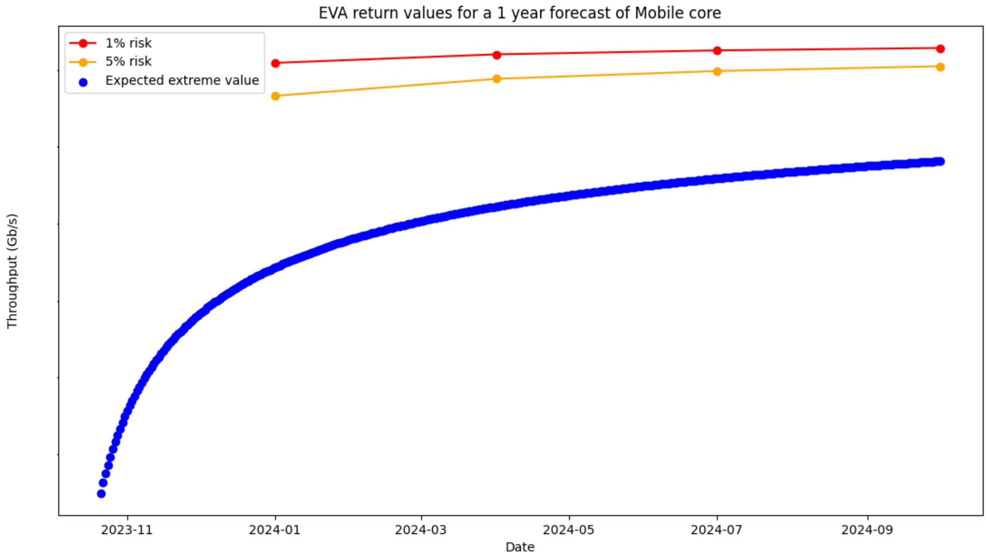


Figure 6.14: The EVA return levels of Mobile Core.

The return levels again represent the deviations from the 31-day moving average and thus represent the predictions of the expected deviations from the average throughput for the next four quarters. The return levels are added to the forecast of the 31-day moving average, to obtain the absolute expected traffic peaks. Figure 6.15 depicts the absolute results of the prediction framework.

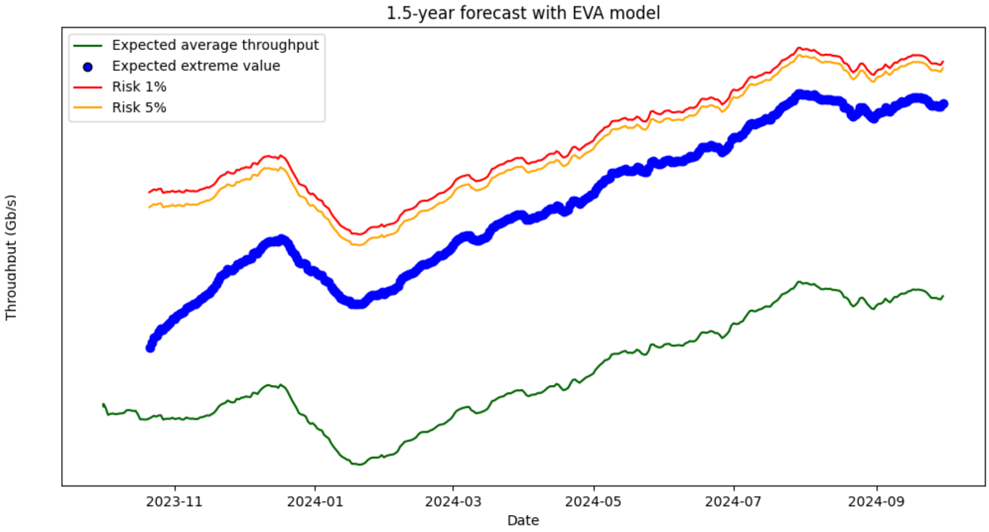


Figure 6.15: The results of the EVA model with risk levels for Mobile Core.

The EVA model calculates the maximum reached throughput per quarter based on the calculated return levels.

The prediction framework can accurately be implemented for the Mobile Core time series by the adaptation of the forecast horizon of six quarters to one year. Additionally, this framework is applicable for the Wholesale time series, to provide estimates on the maximum throughput and risk values for the next six quarters. However, to apply the framework to the Data Center and Video Data Center domains, a tailored adaptation of the framework is required due to the limited time series data available. It is imperative to accumulate more measurements over time until a substantial dataset is acquired, that allows for accurate throughput predictions for one year to six quarters in advance.

7 Implementation for real-time data

In this chapter, the implementation in DataIku is elaborated on and the steps involved to obtain a real-time operational model for capacity management purposes.

7.1. DataIku pipeline

The final prediction framework with EVA and the DeepAR models have been designed in DataIku for real-time further work. Although the DeepAR model did not yield accurate predictions, the algorithm has shown its potential for further research and therefore was integrated into the pipeline created in DataIku. A flow was created where the models have been implemented to enable their usage in capacity planning tools. This flow is initiated daily to update the historical dataset with the maximum peak value from the previous day, to ensure a continuously updated daily forecast.

A pipeline has been created to obtain the daily data from the report that stores the maximum throughput per day. This data is used to create the input dataset for the model. The pipeline is illustrated in Figure 7.1. The report contains the latest daily peak measurements and each day a new peak is added to the dataset, which consists of the historical daily maximum traffic between March 2020 and October 2023. This dataset can then be employed as input for the prediction framework with EVA and as a training dataset for DeepAR.

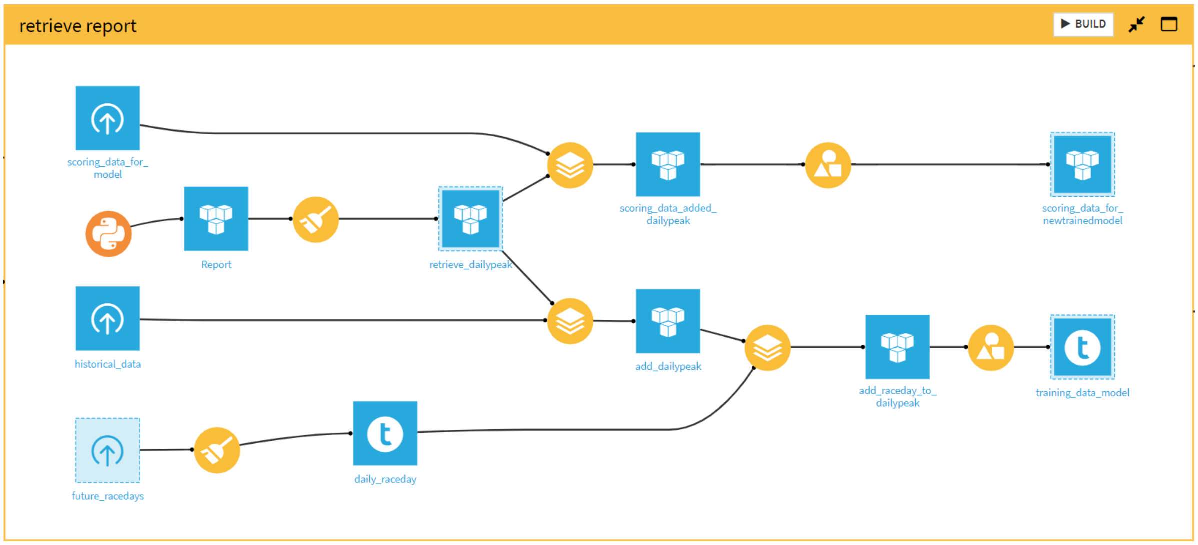


Figure 7.1: The section in the flow that retrieves the report with the daily measurements.

In addition to the retrieval of the network traffic data, the pipeline also incorporates exogenous variables in the pipeline for the DeepAR model. The daily data from the report is integrated into a newly generated dataset, which encompasses not only historical throughput records but also future race days. The model trained on the historical dataset can compute predictions on the throughput with the use of future exogenous variables. As the historical dataset is updated daily, the model is also retrained daily with the new input data. Subsequently, the forecast results of the DeepAR model for the next

year are exported daily to a monitoring tool. Within this tool, the forecast results can be compared to the real-time data and plotted as the future capacity requirements.
The current exogenous dataset consists of the F1 races from 2020 until 2024. This dataset is divided into historical data to train the model and future race days to compute the predictions. In the event of the inclusion of distinct or supplementary events, like football games, a similar approach can be applied that incorporates these specific event occurrences as exogenous variables.

The workflow that contains the prediction framework is shown in Figure 7.2. Firstly, the multiplicative decomposition is performed on the time series to obtain the time series components for the decomposed forecast. Then a seasonal naive model is trained on the decomposed seasonal forecast to obtain the forecast results for the seasonality. This seasonal component is combined with the ARIMA trend forecast and then employed for the Python code written for the EVA estimates.

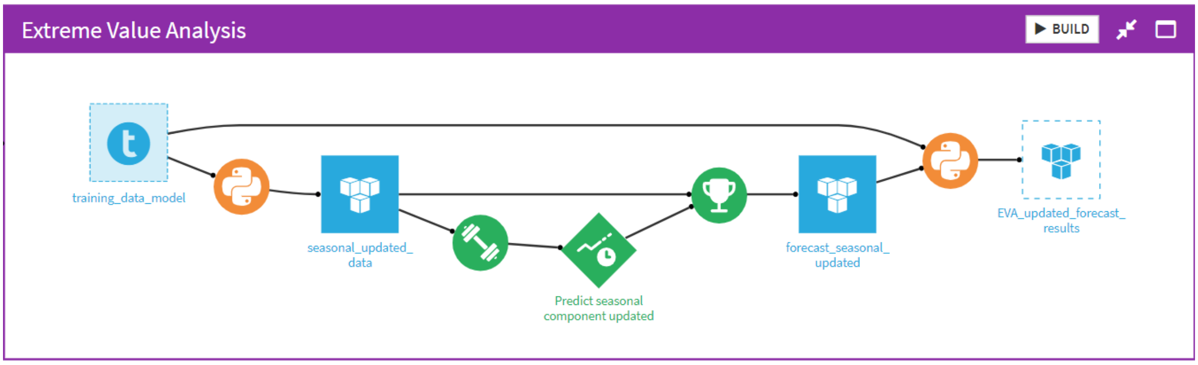

Extreme Value Analysis
BUILD
training_data_model
seasonal_updated_data
Predict seasonal component updated
forecast_seasonal_updated
EVA_updated_forecast_results

Figure 7.2: The part in the flow that executes the EVA model.

The output of this pipeline is the dataset that consists of the results of the EVA model, which gives the absolute maximum expected throughput values per quarter and the throughput values with risk levels of 5% and 1%. Additionally, the EVA results are exported to the monitoring tool and the maximum anticipated throughput values are documented. These results are employed to challenge the manual predictions.

The EVA model that performs the risk computations and combines these with the average throughput forecast, was developed within Visual Studio Code. The Python code written functioned without errors and gave the results as described in the previous chapter. However, to incorporate this framework into DataIku, issues emerged due to the discrepancy in Python versions between the two environments. While Visual Studio Code operates with Python 3.11, DataIku ran on Python 3.7 at the start of this research and currently runs on Python 3.9. This version disparity led to the emergence of new debugging requirements specific to Python 3.9, which led to a significantly time-consuming process. Given the dynamic nature of this code environment, the version will soon transition from 3.9 to 3.11. As such, it is important to carefully consider and address this aspect for the continuity of this research and its models, and take into account version changes that will occur in the future.

To operationalize the Python model within DataIku for real-time usage and business applications in the future, a structured implementation plan was devised. Initially, a concise manual document has been composed, which details the sequential steps within the DataIku workflow and specifies the parameter configurations set for the models. This manual serves as a reference guide, to enable future users of capacity management to seamlessly use and replicate the model and its processes.
Furthermore, a series of collaborative meetings have been conducted, with additional sessions scheduled in the future. These meetings serve a dual purpose; first, to facilitate the seamless handover of the project for real-time integration and second, to pave the way for future research and improvements to the model. Key stakeholders and experts are involved in these discussions, to enable the project to evolve and leverage new data and insights. This will enhance its accuracy and relevance, to address the evolving business needs for a more optimized capacity planning process. This iterative approach is required to update this prediction microservice to the requirements for a dynamic network digital twin.

8 Recommendations for the network

This chapter discusses recommendations for network capacity management, which leverages the insights gained from designing a prediction model for network traffic. It offers opportunities to enhance current measures and leverage data insights. Additionally, it outlines potential new solutions.

8.1. Current transmission measures

The analysis and research performed to design the prediction model for the network domains, have resulted in valuable insights on historical capacity and future demand. These insights contribute to possible solutions for network capacity management, which aims to create a network that can adapt to high demand. Currently, the network design is based on a static network model. For this network, various measures have been implemented to reduce the load on the core network, as previously introduced in Section 2.2.1. These measures are as follows:

1. To bring content that generates high traffic closer to the user in the network, which involves specific television programs and other video streams. Decentralization of content can effectively mitigate the load on the transport core network. To determine the content to decentralize, algorithms identify the frequently watched programs. The placement of the content in the access layer closer to the users, is a viable means to reduce the load on the transport core network. At present, the algorithms only consider popular television programs as content to be decentralized.
2. To employ multicast as a transmission protocol for more content. Multicast replaces a dedicated stream for each user, by one stream that provides identical content to multiple users, thereby saving bandwidth. However, it may not be feasible for all types of content or network infrastructures. The one-to-many communication method employed by multicast makes it unsuitable for applications such as teleconferences and online collaboration. Although multicast supports media streams, its inherent nature makes functionalities such as pausing the stream difficult to achieve. Additionally, the implementation of multicast across different networks is a complex endeavor. Within specific networks, its adoption is valuable for live streams of television programs.

Research has shown that video streams currently cause the most load on the network, specifically via unicast transmission. Unicast streams are employed for content that is streamed to mobile devices and content that is paused or requested at a later time by interactive television applications. Further mitigation can be done, by replacing more unicast streams with multicast streams. Additionally, an analysis on television programs that are paused or requested at a later time is currently not considered for content decentralization. Ratings of television programs and the ratio of live streams versus delayed streams could be used, to understand which streams are most suitable to decentralize. This data should be leveraged to determine which content to decentralize, which can help save bandwidth in the core network.

Video stream services and channels such as Viaplay and ESPN currently offer multicast transmission in the network of KPN. At this moment, multicast transmission is only provided to customers who watch live content via television channels through a STB, as depicted in the current situation in Figure 8.1. Users who stream this content in any other manner currently require unicast connections.

The expansion of multicast beyond STBs to mobile devices can result in significant bandwidth savings. To achieve this, a converter is required that can effectively convert a unicast stream to multicast streams for mobile devices. This solution, as illustrated in solution 1, would require only one dedicated stream to

the core network. Video applications could provide live content via a unicast transmission to a content server in the network, which converts this to a multicast stream to viewers of the content. This would significantly reduce the load by more efficient transmission where only one bitstream is needed, instead of thousands for all users individually.
The next solution, as illustrated in solution 2, would have live video content delivered through a unicast transmission to a decentralized CDN in the access layer. From this point on, the CDN can deliver unicast streams to each user who requests the content. This would decrease the load on the core network and displace the load to the access layer where there is more capacity available.

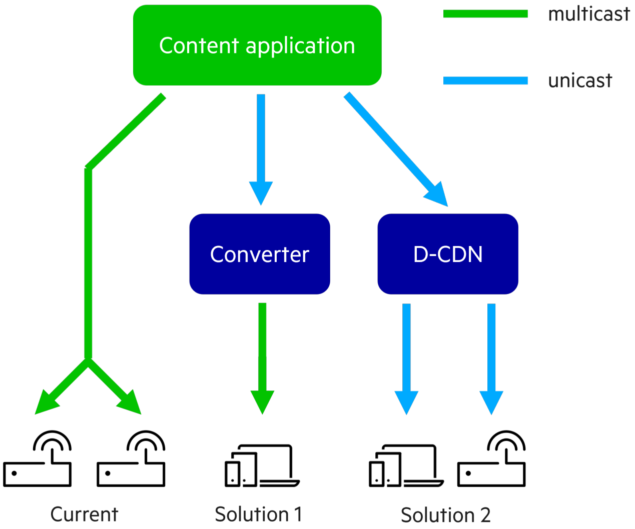


Figure 8.1: The current protocol for non-live streams and the two solutions to stream content to mobile devices and the STB.

To apply these methods, it is necessary to use techniques that ensure low latency. An important requirement to provide live streams of time-sensitive video content to users is low latency. Particularly for live competitions, low latency ensures that users can watch the content in real-time as the events unfold. It is unwanted for users to experience delays in their streams while their neighbors have already reveled the outcome. Currently, the conversion technique needed for solution 1 does not provide a low enough latency and is not implemented. For the second option, decentralization of live content also has to deal with this problem. Content needs to be cached in media segments to be able to decentralize the content. These media segments influence the latency directly. Before the content is cached in various segments, saved and transmitted to the users, a high latency can be expected. Therefore, to implement these techniques it is necessary to use more advanced techniques that ensure a low latency.

8.2. Correlation of peaks

The analysis of time series data is essential to uncover insights into historical usage behavior. It has been shown that there is a notable correlation between traffic peaks and well-known events such as F1 race days, football matches and other popular events. This marks the initial phase of an analysis that links network peaks to specific events. It is crucial to identify the content responsible for each peak and understand its correlation with the magnitude of the traffic peak.
To achieve this, it is necessary to observe and document peaks, to understand and identify events that are responsible for historical extremes. This will result in a labeled dataset that can be used for further analysis. Albeit a time-consuming manual process, this gives valuable insights into the relationship between events and network extremes. A more efficient approach could involve the employment of AI. AI can offer a solution through the automatization of the analysis and correlation of peaks with known events. The initial phase of training AI does involve manual content labeling, but the subsequent stages benefit from an increased efficiency. With the employment of AI, it becomes feasible to discern which events caused specific peaks in historical data. The algorithm has to identify patterns and associate events with network traffic extremes and their magnitudes. If this is automated, a more efficient process

is created than manual analysis.

To help create this foresight, the creation of an event calendar would be preferred. This calendar includes all significant occurrences that will cause high loads and have caused high peaks in the past. This includes for instance sports events, highly anticipated game releases, or other updates. The calendar has to be updated in real-time to obtain all potential situations that can cause a high load on the network. This comprehensive data can then be used to prepare the network for possible high demand moments. When this algorithm is trained and delivers accurate results, more valuable knowledge on the correlations between the magnitude of network peaks and events is created. The input of exogenous variables can be improved by this knowledge, which can contribute to a profound improvement in predictions. A cause-and-effect relationship can enable the anticipation of specific high loads, to plan for future network demands more effectively.

Moreover, it could pave the way for a systematic method to determine dynamically which content is most essential to place in a decentralized CDN or provide with multicast transmission. Furthermore, when certain events take place on the same day or other unexpected situations occur, the network is more prepared to determine what network load will be caused by certain events and can calculate when there is not sufficient capacity. For future network capacity management, it signifies a leap toward an automated capacity planning process. Fewer Excel files have to be used, results are not only more substantiated but are drawn from the measurement of insightful data. It marks the initial phase towards the realization of a dynamic network model. One that can adapt to the larger demands of a network in constant evolution.

8.3. Possible solutions

In the pursuit of the optimization of network performance, several mitigation solutions have been contemplated that are driven by the main objective of capacity improvement. While the feasibility of each solution may vary, they converge on a common goal, the reduction of network load.

- A dynamic CDN that can be implemented by adding extra edge capacity, when high loads due to content streams are expected on the network. This can be deployed on the access level, which shifts a part of the load to a lower level in the network, where more capacity is available per user than in the core network. Popular content can be stored on edge servers, that are placed at strategic locations closer to the user. This solution follows up the unicast to D-CDN unicast transmission in Section 8.1.
- A lease system can be adopted that acquires additional resources without the need for the construction of new physical equipment, which is a costly and time-consuming endeavor. Instead, this concept envisions the use of currently built network modules, where available slots can be temporarily leased to address capacity constraints. In extreme circumstances, these inactive slots could be activated as a contingency measure, for which the company needs to pay when the slots are used. To implement this strategy, negotiations with the vendor of server cabinets are essential to secure the required resources for seamless integration.
- Illegal stream sites can contribute to network congestion due to the uncontrolled distribution of content. To address this issue, a proposed strategy involves blocking traffic generated by unauthorized downloads and streams. The objective is to reduce the load and regain control of the generated traffic within the network infrastructure. However, it is important to consider whether service providers should take the initiative to implement these measures. While the government has previously mandated restrictions on sites like Pirate Bay, it is unclear whether voluntary implementation would be effective in the absence of regulatory directives. Some customers could get dissatisfied with the services of the provider and could switch providers.
- For content delivery, the consideration can be made to reduce the quality of streams of some data content, when the network capacity approaches its capacity limits. This would be a trade-off between the optimization of user experience and network efficiency. The available bandwidth could be adjusted and improved by dynamically scaling the quality of streams in response to real-time network conditions. This has two implications. Firstly, one potential drawback is associated with net neutrality principles. The net neutrality policy advocates equal and unbiased access to

online content for all users. When Internet service providers selectively scale down the bitrate of certain content, it could be perceived as a violation of net neutrality. It introduces differences in types of content and users are not given equal access to available resources. However, in the case that all network resources would fall out, this may be a temporary solution to keep network performance sufficient to provide all services. Secondly, providers may not inherently possess the authority to scale the available bitrate. The control over the bitrate quality typically resides with the applications that deliver the content. This underscores the importance of collaborative efforts between network providers and video content providers to address concerns related to network efficiency without compromising the quality of user experiences.

- Bandwidth throttle for users at access levels can be another potential solution. When situations are expected with high demands, this could prevent potential slowdowns or outages. This could especially be valuable when the fiber rollout is finished and users have a lot of access to capacity, for which the core network is not yet scaled for. Moreover, to distribute bandwidth evenly across users and not have disproportionate shares in resources between users. This can be necessary to mitigate low performance when network congestion arises. Again, careful consideration is necessary to strike a balance that maintains fairness and user satisfaction, which prevents potential concerns related to net neutrality or user experience degradation.
- The implementation of a self-learning algorithm that can dynamically load balance traffic. This approach leverages AI to adaptively distribute network traffic, which optimizes resource utilization based on historical patterns and real-time demands. The algorithm evolves as it receives new data, which continuously refines the knowledge of the intricacies of the network to ensure optimal performance. This would have to be implemented into the digital twin of the network.

Solutions summary

New solutions have been explored in this chapter to alleviate the network of high loads. All solutions are considered by various considerations in Table 8.1. These considerations have been determined through a team analysis with two experts on the network and the traffic loads. Feasibility refers to whether the solution is realistic and achievable. The complexity assesses the level of difficulty associated with the implementation of the solution. The flexibility is described as the possibility of dynamically decreasing the network load. Lastly, customer satisfaction shows the impact on the network performance for the customer.

Table 8.1: Comparison of network optimization solutions.

Solution	**Feasibility**	**Complexity**	**Flexibility**	**Customer Satisfaction**
Transmission measures	Moderate	Moderate	High	High
Dynamic edge CDN	Moderate	Moderate	High	High
Leasing system	Moderate	Moderate	Moderate	High
Illegal streaming blockage	Low	Low	Low	Moderate
Reduce stream quality	Moderate	Moderate	High	Moderate
Bandwidth throttling	High	Moderate	High	Low
Dynamic load balancing	High	High	Moderate	High

These solutions encompass diverse measures for the network, each with distinct purposes. Firstly, the two new approaches for transmission, represent more long-term solutions to provide more efficient transmission and avoid high network loads. This also holds for the implementation of the dynamic CDN. Secondly, the leasing system serves as a short-term and temporary option, which is only feasible in cases where financial arrangements can be negotiated. Finally, the other four solutions entail short-term mitigation measures and focus on the management of the expected load through adjustments in quality, services, and resource allocation, rather than expansion of capacity.

9 Conclusion

This chapter provides a summary of the responses to the research questions and details the methods employed to acquire these answers. Furthermore, it includes a discussion of the study and outlines approaches for future work.

9.1. Conclusion

The objective of this research was to improve network capacity planning for the next year by developing a forecast model that considers extreme values of network traffic. Insights into current and future network traffic have been gained by the performance of a time series analysis, the evaluation of statistical models and a machine learning algorithm. The research focused on network traffic data of the service domains connected to the transport core network of KPN, especially Internet Peering and Mobile Core, for which the time series were derived from the daily maximum throughput values.

The framework was designed for the Internet Peering data, as this time series was deemed the most valuable to analyze. This was due to its highest trend increase and extreme traffic peaks, which presented a challenge for accurate model development. Moreover, the Mobile Core throughput was selected for its valuable usage insights and to assess the designed framework. The data analysis, which involved time series decomposition, uncovered increased non-stationary trends with an annual seasonality. The examination revealed extreme traffic loads that can be correlated with F1 race days and other expected events. This signaled the potential influence of exogenous variables on the accuracy of forecasts.

Two distinct models, SARIMA and DeepAR, were evaluated for their predictive capabilities. SARIMA was unable to discern complex patterns and was limited to weekly seasonality, thus proved unsuitable for the Internet Peering predictions. In contrast, DeepAR demonstrated improved pattern recognition, especially with the incorporation of F1 race days as exogenous variables. Despite the outperformance of SARIMA, DeepAR exhibited challenges in seasonality accuracy and showed difficult interpretability.

To address these limitations, Extreme Value Analysis was introduced as a statistical approach that focuses on extreme values in time series. The forecast model, which combined the decomposed forecasts and EVA, outperformed the other models. EVA effectively considers extreme peak values and provides insights into the maximum expected peaks in the next six quarters. Additionally, it offered information about throughput values associated with specific risk levels.

The substantiated forecasts of the EVA model are compared to the current manual predictions. Both the manual predictions and EVA estimates yielded comparable results. Nonetheless, the EVA model offers more insights into the likelihood of exceeding specific traffic values. This underscores its ability to provide more efficient capacity calculations and enhanced precision. Moreover, the prediction can be automated which enhances the consistency and controllability of the computations. Subsequently, the framework was applied successfully to the Mobile Core time series with an adjusted forecast horizon. This adaptation enables the model to predict the required capacity in the transport core network for all service domains.

Furthermore, the framework with EVA and the DeepAR model have been integrated into the employed business interface for capacity planning purposes. Although DeepAR did not provide predictions of sufficient accuracy, it exhibited promising results and was therefore implemented for further research. An automated pipeline was designed to retrieve the daily maximum throughput for the input of the two models. The models have been integrated into this pipeline, to obtain updated predictions with real-time data. This framework can then be used to compute predictions for every service domain.

Exact estimation of future demand is an intricate challenge. However, peaks from popular events can be expected and for this, to build new capacity is not the only solution to handle the increased load. Various solutions can be employed to scale to the required demand. These solutions can be short-term mitigation solutions or long-term measures that can alleviate high network loads. It is important to consider customer satisfaction for the potential implementation of these solutions. The solutions emphasize the need for careful implementation and integration of dynamic decision-making into the digital twin of the network for sustained effectiveness.

To conclude, the development of a prediction framework with EVA offers an approach to extreme value consideration in network capacity planning with risk estimates. This marks the first step in the automatization of short-term capacity planning to enhance current operational processes. The model employs historical data for a prediction microservice, which can be integrated into a digital twin of the network. This research signifies the initial phase in the design of a dynamic capacity planning model. Ultimately, the goal is to realize a just-in-time capacity strategy. Therefore, precise demand forecasts emerge as a critical prerequisite for optimal resource management and capacity planning in the dynamic environment of telecom networks.

9.2. Discussion

This study has its limitations and uncertainties. These have been divided into three subcategories; the dataset, the employment of SARIMA and DeepAR and lastly the framework with the EVA model.

The network traffic dataset

Firstly, through the inspection of the datasets integral to this study, a few considerations arise. The recorded throughput reveals occasional gaps in the daily measurements due to the inherent instability of the old measurement system. To address this, interpolation techniques were employed to render the datasets suitable for time series prediction. It is imperative, however, to be cautious when interpreting the throughput when the data misses values. A mechanism to check the daily measurements and whether they can still be retrieved is recommended, to ensure the accuracy of the throughput representation on those specific days.

Secondly, the consistency of the dataset is complicated as network architecture changes occurred during the measured years. This shift prompted the transfer of throughput among various network components, particularly for the Data Center and Video Data Center. Consequently, the historical data may exhibit inconsistencies, which poses challenges to the predictability of the time series that are directly affected by the architectural transitions.

Thirdly, the datasets of three service domains have a constraint in length size. Daily measurements for the mobile core domain were not available before October 2021, which resulted in a truncated time series dataset compared to the Internet Peering domain. The granularity of the data before this was the maximum recorded throughput per week, which makes it incompatible with daily data for the creation of a longer dataset.

The architectural change of the newly originated Video Data Center from the Data Center resulted in two time series that are constrained in length. The throughput of the Data Center has a sudden decrease of 50% and the Video Data Center starts to take over this throughput until the two time series stabilize in throughput with their distributed content. These time series are too short for a forecast horizon of one year, so they are unsuitable for a one year prediction with SARIMA and DeepAR. The decision has been made not to use the aggregated time series for the prediction as this is not representative of the actual network infrastructure. Moreover, aggregation of these time series is not allowed as there is no evidence that the daily peaks are recorded at the same time.

Lastly, the current dataset does not contain time stamps. The available information does not indicate at what time the daily peak occurred. Due to this, a more comprehensive analysis of the moments that the network experiences high load and user behavior can not be conducted. The correlation between the peaks and known events is solely based on the date and ratings.

SARIMA and DeepAR

Beyond the intricacies of the dataset, the examination of the seasonality is an important aspect. The data seems to exhibit both weekly and annual seasonality patterns, which are considerations for the development of accurate predictive models. SARIMA models only weekly seasonality while DeepAR can model multiple types of seasonality. However, DeepAR failed to detect the seasonality for the univariate predictions.

Furthermore, in the integration of exogenous variables for the prediction models, challenges arise. In the case of the F1 race days dataset, a scarcity of significant data points is present. This is because only F1 race days are included and thus little additional information is available for the model to be trained on. Moreover, since the impact of F1 on the network began in 2022, data points before this are even less significant. As a result, the 78 race days do not all correlate to significantly higher throughput values. Moreover, the exogenous variables have been implemented as binary inputs and it has not been researched whether a binary input results in the most accurate influence.

The granularity of the model was decided on daily data, as more comprehensive research could be performed for this dataset. The forecast horizon for the univariate prediction models designed with SARIMA and DeepAR was therefore limited, as the time series spanned three and a half years. If weekly throughput values were considered for the design of the model, a longer historical time series was available for the service domains. However, the significant evolution in data usage behavior over the past decade could become a challenge as the usage of longer than three years ago may not be representative of current usage patterns. Nonetheless, it could provide insights into the broader growth trajectory of data usage.

Additionally, the accuracy of the models was evaluated with MAPE scores. This was based on the backtest done on the time series. The requirement for the prediction models was to have a forecast horizon of at least one year in advance. Therefore, all historical data was needed to design the prediction models one year in advance, which meant that the predictions were calculated for the period until October 2024. It was not possible to evaluate the prediction values on realizations of the actual throughput, as this throughput had not occurred yet.

EVA

The final framework consisted of the combined decomposed forecast and the EVA model. The naive seasonal algorithm assumes that the seasonality follows the exact pattern every year. However, in real life, this seasonality could slightly change due to multiple reasons. For instance, summer holidays could commence earlier or later, which would influence the data usage and would result in a change in the seasonal pattern. The naive seasonal algorithm does not take into account these changes. Moreover, the naive seasonal forecast uses the annual seasonality from the decomposition to predict the seasonality. If this decomposition contains some inaccuracies in modeling the seasonality, this would mean that the forecast would repeat these inaccuracies into the future expected seasonality.

The dataset of Δ_i is computed with a moving average of 31 days. It was assumed that this was the most appropriate moving average for this granularity of data. As it defines the dataset of the differences, this moving average must represent the overall average throughput. Whether this window size is the best option, which it seems as the EVA model has a good fit, could still be questioned. Moreover, the moving average uses the throughput values of the past 31 days. It could be argued the average of the days before and after should be taken. However, this was not deemed feasible as the moving average can not be computed with future values when the time series reaches its end.

In addition, the threshold was computed with the IQR equation that is used to define the outliers in a box plot. Various techniques can be used to determine this threshold, but as they all hold subjectivity and uncertainty, it is important to consider the characteristics of the dataset. The three approaches may produce thresholds that include non-extreme values, which leads to a biased model of the tail and does not consider the actual shape of the tail. Moreover, the underlying data distribution is not considered if the sample size is used. The upper whisker threshold was assumed the best fit for the tail of the distribution. However, this threshold selection could have been validated with more methods.

The diagnostic tests conducted on the EVA model indicate that a GPD provides the best theoretical fit to model the extreme values. However, it is uncertain if the actual distribution of the tail of the dataset of Δ_i differs significantly from the assumed GPD distribution. Nevertheless, the Q-Q plots reveal a high correlation coefficient, which suggests a good fit between the model and the data.

Finally, it is not possible to verify the estimated maximum throughput per quarter with the realizations of actual throughput. This is because the model uses all historical data until October 2023 and the next six quarters have not occurred yet. Thus the only way to assess the model is to compare the manual predictions that are used for capacity planning with the new results. This comparison will help to understand what the EVA predictions indicate for capacity planning.

In conclusion, these limitations underscore the complexity of a network traffic dataset that is coherent with the behavior of people. There is no straightforward rigid framework that can be employed to predict data associated with human behavior. Instead, a thorough analysis of the data is imperative to devise such a framework.

9.3. Future work

Multiple recommendations arise from the results of this study to improve the prediction models and extend the research to new insights.

Initially, a pivotal step forward involves a comparison of the prediction models with the actual throughput realizations. This validation process is essential for the measurement of the accuracy and reliability of the models in real-world scenarios, which provides valuable insights for further improvement of the models.

This study has revealed a notable correlation between traffic peaks and major events such as F1 race days and football matches. However, besides the F1 races, the other events have not been introduced as exogenous information for the DeepAR model. The incorporation of these other events and the determination of the optimal input format could enhance the predictive capabilities. For instance, the use of a peak-to-average ratio instead of a binary input can be employed. The calculation of the peak-to-average ratio for known events based on historical data peaks introduces a novel approach to the refinement of exogenous variables.

Furthermore, the creation of an event calendar and testing it as exogenous variables holds promise. This involves the incorporation of diverse events, with a range from sports spectacles such as the Olympic Games to fundraiser events like Giro 555. These events could be extracted from news articles, sports calendars and current proceedings. An event calendar would enrich the model and potentially improve the accuracy of extreme peaks.

In addition, the data collected during the COVID-19 lockdown period is reflective of a unique time when people were restricted to their homes. Although this data is more relevant to present-day behavior as hybrid working has become a new trend, people are no longer required to stay at home. People have resumed their usual routines, which include work at the office or educational institutes, as well as attending events. Furthermore, the high loads caused by press conferences, prevalent in the lockdown, are unlikely to recur. These shifts in usage behavior are inevitable and are a challenge for prediction models. Nonetheless, the usage of this data is valuable, given the dynamic nature of network traffic which continuously evolves with unexpected new trends, such as game releases or advancements in

AR/VR technology and IoT. This underscores the importance of prediction models that exhibit resilience to changes in user behavior, which has to be the focus of future tests.
Also, time stamps of network peaks should be recorded. Events that occur in the evening, such as some F1 races, could be assigned a higher factor for exogenous input than events that occur by day, as fewer people are expected to have time to watch then. If higher factors are assigned to these events, the greater impact on the network load in peak hours is acknowledged. Additionally, it is essential to understand the distribution of load over time, particularly in cases of concurrent events, for the optimization of network resource allocation.

Furthermore, it is recommended to enhance the data measurement pipeline by retaining daily traffic profiles alongside maximum values. This approach offers a more detailed perspective on network performance and helps identify usage patterns. It also enables a dynamic prediction model for resource optimization, which could lead to cost savings and improved network efficiency. Moreover, the usage of daily traffic profiles can extend beyond capacity planning. For instance, they can be valuable for anomaly detection, as they facilitate the identification of anomalies or irregularities in network usage more effectively. Sudden drops or peaks in the throughput can be indicative of network issues or emerging trends. This allows for quicker responses and better troubleshooting.

This research demonstrated that the scope of data analysis and its applications is contingent upon factors such as data storage and granularity. Increased data storage facilitates additional possibilities for data analysis and deeper insights. Given the evolving landscape of AI, machine learning and other emerging technologies, there is a need to reassess data pipelines and make informed decisions on the retention or elimination of specific data. This is essential for the automatization of operational processes and reconsideration of strategies.

Multivariate time series forecasts are the next step towards more comprehensive models. The DeepAR model can learn from various time streams and should take into account external factors like weather or concurrent events. For instance, the usage of Netflix tends to decrease when the weather is pleasant, a pattern that could be captured by multivariate prediction. Additionally, Netflix usage may decline when a larger audience is engaged in live television broadcasts. The adoption of a multivariate approach is essential for the model to recognize and adapt to seasonality patterns more accurately.

A new addition to a dynamic model could be to translate the percentage of the load on resources into costs. Currently, there are no insights on the overhead expenses. Understanding the cost implications of network utilization provides essential insights to optimize resource management. A more efficient and economically informed model could be created with this knowledge.

These proposals aim to enhance the predictive capabilities of the researched DeepAR and EVA models. The objective to improve network capacity planning can be extended by the improvement of data insights and exogenous information, for real-world applicability.

References

[1] *ADL Data Growth Europe 2023*. https://www.adlittle.com/sites/default/files/reports/ADL_Data_growth_Europe_2023.pdf.

[2] Rituparna De', Niharika Pandey, and Arpan Pal. "Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice". In: *International Journal of Information Management* 55 (2020), p. 102171. DOI: 10.1016/j.ijinfomgt.2020.102171.

[3] Niombo Lomba, Lenka Jančová, and Meenakshi Fernandes. *EPRS | European Parliamentary Research Service*. Tech. rep. PE 699.475. European Added Value Unit, Jan. 2022.

[4] Carlos Alberto et al. "Capacity Planning In a Telecommunications Network: A Case Study". In: *The International Journal of Industrial Engineering: Theory, Applications and Practice* 16 (June 2009).

[5] Josep Ferrandiz and Alex Gilgur. "Capacity Planning for QoS". In: *The Journal of Computer Resource Management* Winter 2014 (Jan. 2014), p. 15.

[6] KPN. *KPN at a Glance*. Accessed: [16/11/2023]. 2023. URL: https://www.overons.kpn/en/the-company/kpn-at-a-glance.

[7] KPN. *Data Traffic on KPN Network Grew 40% During Corona*. 2021. URL: https://www.overons.kpn/nieuws/en/data-traffic-on-kpn-network-grew-40-during-corona/.

[8] KPN. *Dutch People Abroad Follow Home Race in Zandvoort Closely*. enter year. URL: https://www.overons.kpn/nieuws/en/dutch-people-abroad-follow-home-race-in-zandvoort-closely/.

[9] David Jones et al. "Characterising the Digital Twin: A systematic literature review". In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52. ISSN: 1755-5817. DOI: https://doi.org/10.1016/j.cirpj.2020.02.002. URL: https://www.sciencedirect.com/science/article/pii/S1755581720300110.

[10] Dataiku, Inc. *Dataiku*. Version 12. 2023. URL: https://www.dataiku.com/.

[11] Carlos Alberto et al. "Capacity Planning In a Telecommunications Network: A Case Study". In: *The International Journal of Industrial Engineering: Theory, Applications and Practice* 16 (June 2009).

[12] Mohammad Noormohammadpour and Cauligi S. Raghavendra. "Datacenter Traffic Control: Understanding Techniques and Tradeoffs". In: *IEEE Communications Surveys & Tutorials* 20.2 (2018), pp. 1492–1525. DOI: 10.1109/COMST.2017.2782753.

[13] Igor Tomic, Eoin Bleakley, and Predrag Ivanis. "Predictive Capacity Planning for Mobile Networks&mdash;ML Supported Prediction of Network Performance and User Experience Evolution". In: *Electronics* 11.4 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11040626. URL: https://www.mdpi.com/2079-9292/11/4/626.

[14] Edward Oughton et al. "Assessing the capacity, coverage and cost of 5G infrastructure strategies: Analysis of The Netherlands". In: *Telematics and Informatics* (Apr. 2019). DOI: 10.1016/j.tele.2019.01.003.

[15] Hira Zahid et al. "Big data analytics in telecommunications: literature review and architecture recommendations". In: *IEEE/CAA Journal of Automatica Sinica* 7.1 (2020), pp. 18–38. DOI: 10.1109/JAS.2019.1911795.

[16] Bo Ma, Weisi Guo, and Jie Zhang. "A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning". In: *IEEE Access* 8 (2020), pp. 35606–35637. DOI: 10.1109/ACCESS.2020.2975004.

[17] Atif Mahmood et al. “Capacity and Frequency Optimization of Wireless Backhaul Network Using Traffic Forecasting”. In: *IEEE Access* 8 (2020), pp. 23264–23276. DOI: 10.1109/ACCESS.2020.2970224.

[18] Gabriel O. Ferreira et al. “Forecasting Network Traffic: A Survey and Tutorial With Open-Source Comparative Evaluation”. In: *IEEE Access* 11 (2023), pp. 6018–6044. DOI: 10.1109/ACCESS.2023.3236261.

[19] Zhiyong Xu et al. “Efficient Prediction of Network Traffic for Real-Time Applications”. In: *Journal of Computer Networks and Communications* 2019 (2019), p. 4067135. DOI: 10.1155/2019/4067135. URL: https://doi.org/10.1155/2019/4067135.

[20] Ola Surakhi et al. “Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm”. In: *Electronics* 10 (Oct. 2021). DOI: 10.3390/electronics10202518.

[21] Purnawansyah Purnawansyah et al. “Network Traffic Time Series Performance Analysis Using Statistical Methods”. In: *Knowledge Engineering and Data Science* 1 (Dec. 2017), p. 1. DOI: 10.17977/um018v1i12018p1-7.

[22] Sima Siami Namini, Neda Tavakoli, and Akbar Siami Namin. “A Comparison of ARIMA and LSTM in Forecasting Time Series”. In: Dec. 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[23] Vijay Kotu and Bala Deshpande. “Chapter 12 - Time Series Forecasting”. In: *Data Science (Second Edition)*. Ed. by Vijay Kotu and Bala Deshpande. Second Edition. Morgan Kaufmann, 2019, pp. 395–445. ISBN: 978-0-12-814761-0. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00012-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128147610000125.

[24] G. Box, Gwilym Jenkins, and C. Reinsel. “Time series analysis: forecasting and control (third ed”. In: *Time Series Analysis: Forecasting and Control: Fourth Edition* (May 2013). DOI: 10.1002/9781118619193.

[25] Xin Dong, Wentao Fan, and Jun Gu. “Predicting LTE Throughput Using Traffic Time Series”. In: *ZTE communications* 13 (2015), pp. 61–64. URL: https://api.semanticscholar.org/CorpusID:113040253.

[26] Yanhua Yu et al. “Network Traffic Prediction and Result Analysis Based on Seasonal ARIMA and Correlation Coefficient”. In: *2010 International Conference on Intelligent System Design and Engineering Application*. Vol. 1. 2010, pp. 980–983. DOI: 10.1109/ISDEA.2010.335.

[27] Yantai Shu et al. “Wireless Traffic Modeling and Prediction Using Seasonal ARIMA Models”. In: *IEICE Transactions on Communications* E88B (Jan. 2003). DOI: 10.1093/ietcom/e88-b.10.3992.

[28] Aileen Nielsen. *Practical time series analysis: Prediction with statistics and machine learning*. O’Reilly Media, 2019.

[29] Jatinder Kaur, Kulwinder Parmar, and Sarbjit Singh. “Autoregressive models in environmental forecasting time series: a theoretical and application review”. In: *Environmental Science and Pollution Research* 30 (Jan. 2023), pp. 1–25. DOI: 10.1007/s11356-023-25148-9.

[30] Samuel Medhn et al. “Mobile data traffic forecasting in UMTS networks based on SARIMA model: The case of Addis Ababa, Ethiopia”. In: *2017 IEEE AFRICON*. 2017, pp. 285–290. DOI: 10.1109/AFRCON.2017.8095496.

[31] Fahad Radhi Alharbi and Denes Csala. “A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach”. In: *Inventions* 7.4 (2022). ISSN: 2411-5134. DOI: 10.3390/inventions7040094. URL: https://www.mdpi.com/2411-5134/7/4/94.

[32] Thanh Tran et al. “A Multiplicative Seasonal ARIMA/GARCH Model in EVN Traffic Prediction”. In: *International Journal of Communications, Network and System Sciences* 08 (Jan. 2015), pp. 43–49. DOI: 10.4236/ijcns.2015.84005.

[33] Yunxue Gao et al. “Short Term Prediction Models of Mobile Network Traffic Based on Time Series Analysis”. In: Jan. 2018, pp. 205–211. ISBN: 978-3-319-73563-4. DOI: 10.1007/978-3-319-73564-1_20.

[34] Mircea Eugen Dodan, Quoc-Tuan Vien, and Tuan Thanh Nguyen. “Internet Traffic Prediction Using Recurrent Neural Networks”. In: *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 9.4 (Sept. 2022). DOI: `10.4108/eetinis.v9i4.1415`.

[35] Imad Alawe et al. “Improving Traffic Forecasting for 5G Core Network Scalability: A Machine Learning Approach”. In: *IEEE Network* 32 (Nov. 2018), pp. 42–49. DOI: `10.1109/MNET.2018.1800104`.

[36] Davide Andreoletti et al. “Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks”. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, pp. 246–251. DOI: `10.1109/INFCOMW.2019.8845132`.

[37] Mousa Alizadeh et al. “Network Traffic Forecasting Based on Fixed Telecommunication Data Using Deep Learning”. In: Dec. 2020, pp. 1–7. DOI: `10.1109/ICSPIS51611.2020.9349573`.

[38] Chengsheng Pan et al. “Network Traffic Prediction Incorporating Prior Knowledge for an Intelligent Network”. In: *Sensors* 22 (Mar. 2022), p. 2674. DOI: `10.3390/s22072674`.

[39] Nipun Ramakrishnan and Tarun Soni. “Network Traffic Prediction Using Recurrent Neural Networks”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 187–193. DOI: `10.1109/ICMLA.2018.00035`.

[40] Quang Hung Do et al. “Prediction of Data Traffic in Telecom Networks based on Deep Neural Networks”. In: *Journal of Computer Science* 16 (Sept. 2020), pp. 1268–1277. DOI: `10.3844/jcssp.2020.1268.1277`.

[41] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. “A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences”. In: *Intelligent Computing*. Springer Nature Switzerland, 2023, pp. 771–785. ISBN: 9783031379635. DOI: `10.1007/978-3-031-37963-5_53`. URL: `http://dx.doi.org/10.1007/978-3-031-37963-5_53`.

[42] Michael Franklin Mbouopda et al. “Experimental Study of Time Series Forecasting Methods for Groundwater Level Prediction”. In: *Advanced Analytics and Learning on Temporal Data*. Ed. by Thomas Guyet et al. Cham: Springer International Publishing, 2023, pp. 34–49. ISBN: 978-3-031-24378-3.

[43] Aleksei Mashlakov et al. “Assessing the performance of deep learning models for multivariate probabilistic energy forecasting”. In: *Applied Energy* 285 (2021), p. 116405. ISSN: 0306-2619. DOI: `https://doi.org/10.1016/j.apenergy.2020.116405`. URL: `https://www.sciencedirect.com/science/article/pii/S0306261920317748`.

[44] David Salinas et al. “DeepAR: Probabilistic forecasting with autoregressive recurrent networks”. In: *International Journal of Forecasting* 36.3 (2020), pp. 1181–1191. ISSN: 0169-2070. DOI: `https://doi.org/10.1016/j.ijforecast.2019.07.001`. URL: `https://www.sciencedirect.com/science/article/pii/S0169207019301888`.

[45] Jiachen Zhang et al. “Base Station Network Traffic Prediction Approach Based on LMA -DeepAR”. In: Apr. 2021, pp. 473–479. DOI: `10.1109/ICCCS52626.2021.9449212`.

[46] Carolina Gijón et al. “Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series”. In: *Electronics* 10.10 (2021). ISSN: 2079-9292. DOI: `10.3390/electronics10101151`. URL: `https://www.mdpi.com/2079-9292/10/10/1151`.

[47] Holger Kantz et al. “Dynamical Interpretation of Extreme Events: Predictability and Predictions”. In: Jan. 2006, pp. 69–93. ISBN: 978-3-540-28610-3. DOI: `10.1007/3-540-28611-X_4`.

[48] Fengli Xu et al. “Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach”. In: *IEEE Transactions on Services Computing* 9.5 (2016), pp. 796–805. DOI: `10.1109/TSC.2016.2599878`.

[49] 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)*. 2021, pp. 1–6. DOI: `10.1109/VTC2021-Spring51267.2021.9449040`.

[50] Daizong Ding et al. “Modeling Extreme Events in Time Series Prediction”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1114–1122. ISBN: 9781450362016. DOI: `10.1145/3292500.3330896`. URL: `https://doi-org.tudelft.idm.oclc.org/10.1145/3292500.3330896`.

[51] Myriam Garrido, Yves Deville, and Pascal Lezaud. “Corrective to the article : Extreme Value Analysis - an Introduction Journal de la SFdS Vol. 154 No2, 66-97”. In: *Journal de la Societe Française de Statistique* 58.3 (2017), pp. 27–28. URL: https://hal.inrae.fr/hal-02619684.

[52] B. Finkenstädt and Holger Rootzén. *Extreme values in finance, telecommunications, and the environment*. Jan. 2003, pp. 1–389.

[53] Manfred Gilli and Evis këllezi. “An Application of Extreme Value Theory for Measuring Financial Risk”. In: *Computational Economics* 27 (Feb. 2006), pp. 207–228. DOI: 10.1007/s10614-006-9025-7.

[54] Jelena Jockovic. “Quantile estimation for the generalized pareto distribution with application to finance”. In: *Yugoslav Journal of Operations Research* 22 (Jan. 2012). DOI: 10.2298/YJOR110308013J.

[55] Zoi Tsourti and John Panaretos. “Extreme-value analysis of teletraffic data”. In: *Computational Statistics & Data Analysis* 45.1 (2004). Computer Security and Statistics, pp. 85–103. ISSN: 0167-9473. DOI: https://doi.org/10.1016/S0167-9473(03)00116-6. URL: https://www.sciencedirect.com/science/article/pii/S0167947303001166.

[56] M. Uchida. “Traffic data analysis based on extreme value theory and its applications”. In: *IEEE Global Telecommunications Conference, 2004. GLOBECOM ’04.* Vol. 3. 2004, 1418–1424 Vol.3. DOI: 10.1109/GLOCOM.2004.1378217.

[57] Chunfeng Liu et al. “Application of Extreme Value Theory to the Analysis of Wireless Network Traffic”. In: July 2007, pp. 486–491. ISBN: 1-4244-0353-7. DOI: 10.1109/ICC.2007.86.

[58] Rob Hyndman and Andrey Kostenko. “Minimum Sample Size Requirements for Seasonal Forecasting Models”. In: *Foresight: The International Journal of Applied Forecasting* 6 (Feb. 2007), pp. 12–15.

[59] Anja Feldmann et al. “Implications of the COVID-19 Pandemic on the Internet Traffic”. In: *Broadband Coverage in Germany; 15th ITG-Symposium*. 2021, pp. 1–5.

[60] Mathieu Lepot, Jean-Baptiste Aubin, and François H.L.R. Clemens. “Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment”. In: *Water* 9.10 (2017). ISSN: 2073-4441. DOI: 10.3390/w9100796. URL: https://www.mdpi.com/2073-4441/9/10/796.

[61] Aistis Raudys and Židrina PABARŠKAITĖ. “Optimising the smoothness and accuracy of moving average for stock price data”. In: *Technological and Economic Development of Economy* 24 (May 2018), pp. 984–1003. DOI: 10.3846/20294913.2016.1216906.

[62] V. Chaurasia and S. Pal. “Application of machine learning time series analysis for prediction COVID-19 pandemic”. In: *Research on Biomedical Engineering* 38.1 (2022), pp. 35–47. DOI: 10.1007/s42600-020-00105-4. URL: https://doi.org/10.1007/s42600-020-00105-4.

[63] “Introduction”. In: *Introduction to Time Series and Forecasting*. Ed. by Peter J. Brockwell and Richard A. Davis. New York, NY: Springer New York, 2002, pp. 1–44. ISBN: 978-0-387-21657-7. DOI: 10.1007/0-387-21657-X_1. URL: https://doi.org/10.1007/0-387-21657-X_1.

[64] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[65] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 2nd ed. Melbourne, Australia: OTexts, 2018.

[66] David A Dickey and Wayne A Fuller. “Distribution of the estimators for autoregressive time series with a unit root”. In: *Journal of the American statistical association* 74.366a (1979), pp. 427–431.

[67] Denis Kwiatkowski et al. “Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?” In: *Journal of Econometrics* 54.1 (1992), pp. 159–178. ISSN: 0304-4076. DOI: https://doi.org/10.1016/0304-4076(92)90104-Y. URL: https://www.sciencedirect.com/science/article/pii/030440769290104Y.

[68] G.Peter Zhang. “Time series forecasting using a hybrid ARIMA and neural network model”. In: *Neurocomputing* 50 (2003), pp. 159–175. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(01)00702-0. URL: https://www.sciencedirect.com/science/article/pii/S0925231201007020.

[69] Manfred Deistler and Wolfgang Scherrer. “Models with Exogenous Variables”. In: *Time Series Models*. Cham: Springer International Publishing, 2022, pp. 155–166. ISBN: 978-3-031-13213-1. DOI: 10.1007/978-3-031-13213-1_8. URL: https://doi.org/10.1007/978-3-031-13213-1_8.

[70] Rob J. Hyndman and Yeasmin Khandakar. “Automatic Time Series Forecasting: The forecast Package for R”. In: *Journal of Statistical Software* 27.3 (2008), pp. 1–22. DOI: 10.18637/jss.v027.i03. URL: https://www.jstatsoft.org/index.php/jss/article/view/v027i03.

[71] Amazon Web Services, Inc. *Amazon SageMaker DeepAR Documentation*. [Online; accessed September 2023]. 2023. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html.

[72] Sonia Benito, Carmen Lopez, and MªÁngeles Navarro. “Assessing the importance of the choice threshold in quantifying market risk under the POT approach”. In: *Risk Management* 25 (Jan. 2023), p. 1. DOI: 10.1057/s41283-022-00106-w.

[73] Stuart Coles. “Threshold Models”. In: *An Introduction to Statistical Modeling of Extreme Values*. London: Springer London, 2001, pp. 74–91. ISBN: 978-1-4471-3675-0. DOI: 10.1007/978-1-4471-3675-0_4. URL: https://doi.org/10.1007/978-1-4471-3675-0_4.

[74] Myriam Garrido and Pascal Lezaud. “Extreme Value Analysis: an Introduction”. In: *Journal de la Societe Francaise de Statistique* 154.2 (2013), pp. 66–97. URL: https://hal.archives-ouvertes.fr/ffhal-00917995.

[75] Vladimir O. Andreev et al. “Extreme Value Theory and Peaks Over Threshold Model in the Russian Stock Market”. In: *Journal of Siberian Federal University. Engineering & Technologies* 1.5 (2012), pp. 111–121.

[76] Mikhail Makarov. “Applications of exact extreme value theorem”. In: *Journal of Operational Risk* 2 (Mar. 2007), pp. 115–120. DOI: 10.21314/JOP.2007.024.

[77] Arnaud Clément-Grandcourt and Hervé Fraysse. “3 - How to Use These Scenarios for Asset Management?” In: *Hazardous Forecasts and Crisis Scenario Generator*. Ed. by Arnaud Clément-Grandcourt and Hervé Fraysse. Elsevier, 2015, pp. 75–126. ISBN: 978-1-78548-028-7. DOI: https://doi.org/10.1016/B978-1-78548-028-7.50003-5. URL: https://www.sciencedirect.com/science/article/pii/B9781785480287500035.

[78] Sergio Juárez and William Schucany. “Robust and Efficient Estimation for the Generalized Pareto Distribution”. In: *Extremes* 7 (Sept. 2004), pp. 237–251. DOI: 10.1007/s10687-005-6475-6.

[79] Sebastián Solari et al. “Peaks Over Threshold (POT): A methodology for automatic threshold estimation using goodness of fit p-value”. In: *Water Resources Research* 53.4 (2017), pp. 2833–2849. DOI: https://doi.org/10.1002/2016WR019426. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2016WR019426. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016WR019426.

[80] Bikramjit Das and Souvik Ghosh. “Detecting tail behavior: mean excess plots with confidence bounds”. In: *Extremes* 19 (June 2016). DOI: 10.1007/s10687-015-0238-9.

[81] Souvik Ghosh and Sidney Resnick. “A discussion on mean excess plots”. In: *Stochastic Processes and their Applications* 120.8 (2010), pp. 1492–1517. ISSN: 0304-4149. DOI: https://doi.org/10.1016/j.spa.2010.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0304414910001079.

[82] Frank E. Grubbs. “Procedures for Detecting Outlying Observations in Samples”. In: *Technometrics* 11.1 (1969), pp. 1–21. DOI: 10.1080/00401706.1969.10490657.

[83] Laura Schneider, Andrea Krajina, and Tatyana Krivobokova. “Threshold selection in univariate extreme value analysis”. In: *Extremes* 24 (Dec. 2021). DOI: 10.1007/s10687-021-00405-7.

[84] M. J. Hall et al. “The construction of confidence intervals for frequency analysis using resampling techniques”. In: *Hydrology and Earth System Sciences* 8.2 (2004), pp. 235–246. DOI: 10.5194/hess-8-235-2004. URL: https://hess.copernicus.org/articles/8/235/2004/.

[85] I. Osetinsky-Tzidaki and E. Fredj. “The 50- and 100-year Exceedance Probabilities as New and Convenient Statistics for a Frequency Analysis of Extreme Events: An Example of Extreme Precipitation in Israel”. In: *Water* 15 (2023), p. 44.

[86] George Bocharov. *Pyextremes Package*. 2023. URL: https://georgebv.github.io/pyextremes/.

[87] Ed Mackay and Philip Jonathan. "Sampling properties and empirical estimates of extreme events". In: *Ocean Engineering* 239 (2021), p. 109791. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2021.109791. URL: https://www.sciencedirect.com/science/article/pii/S0029801821011549.

[88] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2019.132306. URL: https://www.sciencedirect.com/science/article/pii/S0167278919305974.

[89] Joos Korstanje. *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*. Apress, 2021. ISBN: 9781484271506.

List of Figures

List of Tables

A
Time series analysis

A.1. Stationarity tests

Two tests have been conducted on the original time series from 09/03/2020 until 01/10/2023 to verify that the time series are non-stationary, as has been empirically determined. The results of these tests can be seen in Table A.1.

ADF Test	
ADF Statistic:	-0.4555151710733376
p-value:	0.9004090032725548
Critical Values:	
1%	-3.435469111362934
5%	-2.8638006501960755
10%	-2.567973589477539
KPSS Test	
KPSS Statistic:	5.387415
p-value:	0.010000
Critical Values:	
10%	0.347
5%	0.463
2.5%	0.574
1%	0.739

Table A.1: ADF and KPSS Test Results

A.2. Autocorrelation

To estimate the orders of the ARIMA model, the ACF and PACF plots for the Internet Peering model were performed. This provides an initial reference for determining the appropriate lags to incorporate. Figure A.1 shows the results of these functions. A seasonal cycle of 7 periods can be seen in the ACF plot. This indicates that there is a repeating pattern in the dataset that occurs every seven days. This recurring cycle suggests that there is a weekly seasonality within the time series.

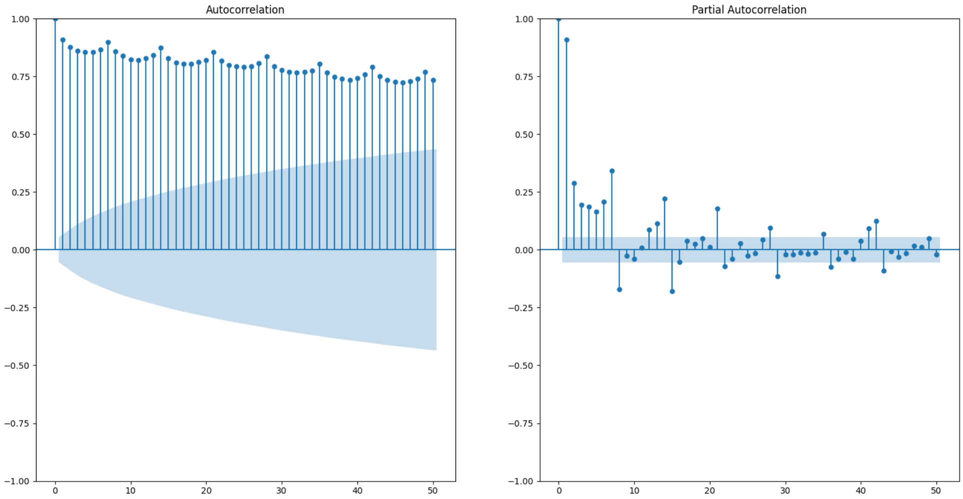


Figure A.1: The ACF and the PACF plots of the Internet Peering time series

The AFC plot shows a gradual decrease as the lags increase, which suggests a slow decay of correlation and indicates that the time series is not stationary as shown above. The PACF plot tails off after the first lag, which suggests that an autoregressive term of 1 could be a fit so AR(1) will be used for ARIMA.

Figure A.2 shows the test to determine the order of differencing needed for the Internet peering time series. After the first order of differencing, the ACF plot significantly improves as it shows a quick drop-off, which is a characteristic of a stationary series. The second order differencing plot does not show a substantial improvement over the 1st order and might suggest over-differencing as indicated by the alternating positive and negative lags. This can lead to loss of information and more model complexity. Therefore, the time series is stationary after the first order of differencing and therefore $d = 1$ can be used for the ARIMA model.

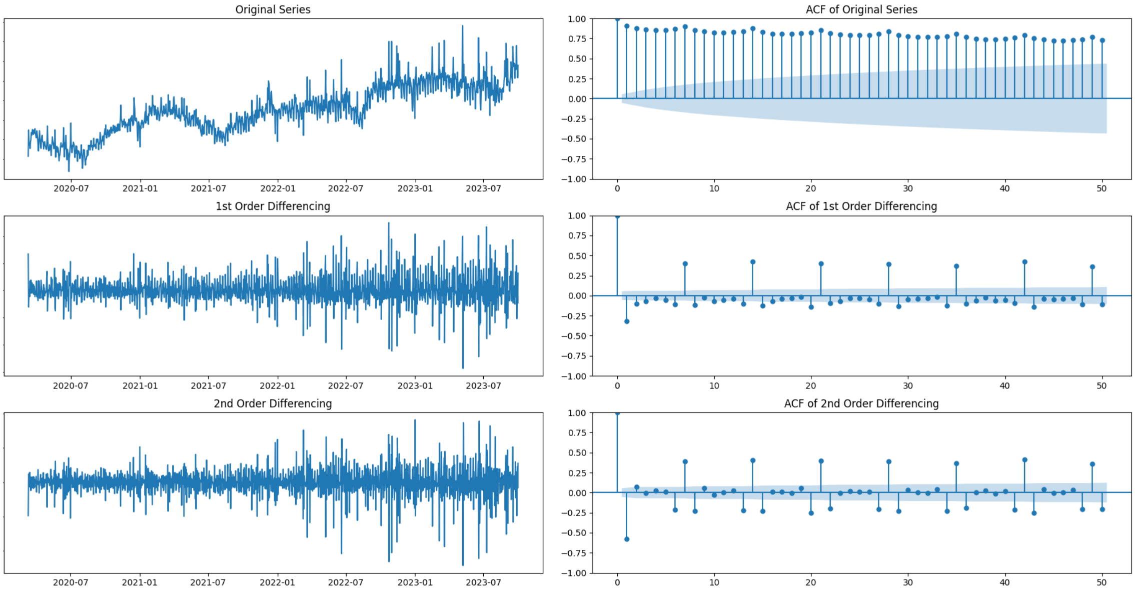


Figure A.2: Differencing of the Internet Peering time series

The ACF plot after the first order of differencing displays a significant spike at lag 1, followed by autocorrelation values within the confidence bounds for subsequent lags. This suggests that an MA(1) model might be appropriate, as the series appears to be influenced by the error term of the immediate past value. The lack of further spikes indicates that an ARIMA model with a component of order 1 could be a good fit.

A.3. Throughput per day of the week

Figure A.3 demonstrates again that the traffic generated by users on Tuesdays has the highest overall throughput, but during the second quarter of 2022, this changed to Sundays. On Saturdays, the load on the network seems to be the lowest. People are most likely to do trips or enjoy activities on Saturdays, which explains this traffic usage. The network traffic on all days has shown to be consistent during the years, except for traffic on Sundays. Moreover, there appears to be a change in the day for which the second most traffic is generated. The moving average on Wednesday shows to surpass the traffic on Tuesdays. Both add to a significant change in usage behavior during the past 3,5 years, which can cause complexity for time series prediction models.

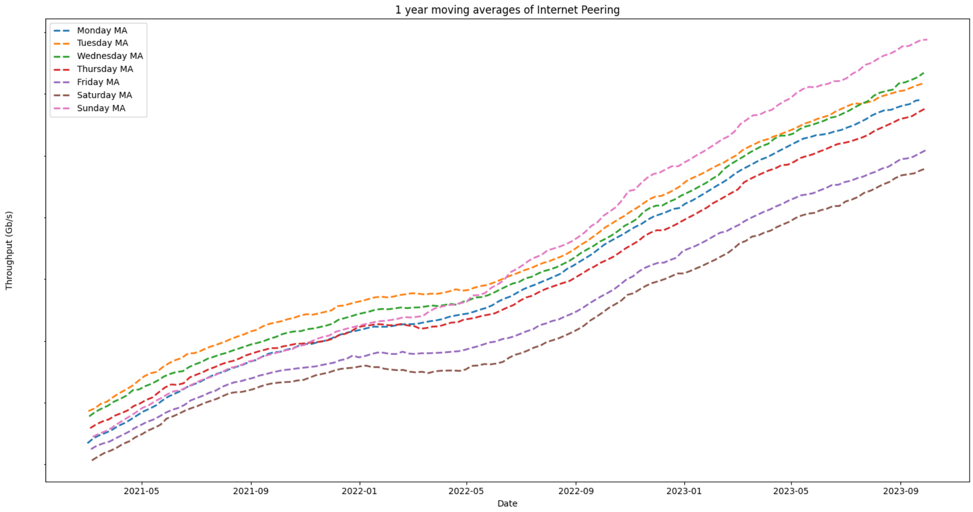


Figure A.3: The one year moving average of Internet Peering for every day of the week.

B Time series models

B.1. SARIMA

The ARIMA model combines AR and MA processes to create a composite model for time series data. In the AR component, past values influence the current value. This is defined as described in the following equation:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + e_t$$

For the MA part, it considers past error terms which is described as:

$$Y_t = \mu + e_t + \sum_{i=1}^{q} \theta_i e_{t-i}$$

When these two components are combined, the following model is obtained that generates the forecast [68]:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \ldots - \theta_q e_{t-q} + e_t$$

Here Y_t represents the actual values at time t, c is the constant term, and e_t the error terms at time t. The model parameters consist of the AR parameter ϕ_i, for $i = 1, 2, ..., p$ and the MA parameter θ_j, for $j = 1, 2, ..., q$. The integers p and q are the model orders.

The SARIMA model is denoted by ARIMA(p,d,q)(P,D,Q)[m], where each parameter plays a distinct role in the formulation of the model. Here d indicates the degree of differencing required to make the series stationary and m the season length, thus the number of periods in each season. The selection of these parameters is based on tests for stationarity and autocorrelation within the data, to ensure the model is well-suited to capture the underlying patterns of the time series.

Figure B.1 shows the results of the first model. The historical time series is plotted, as well as the backtest and the forecast. Due to the plotting settings of DataIku, the x-axis does not show all of the historical data that was used as an input. Again the backtest is shown on the test set of the time series, which is identical to that of the second model als both ARIMA approaches resulted in the same model.

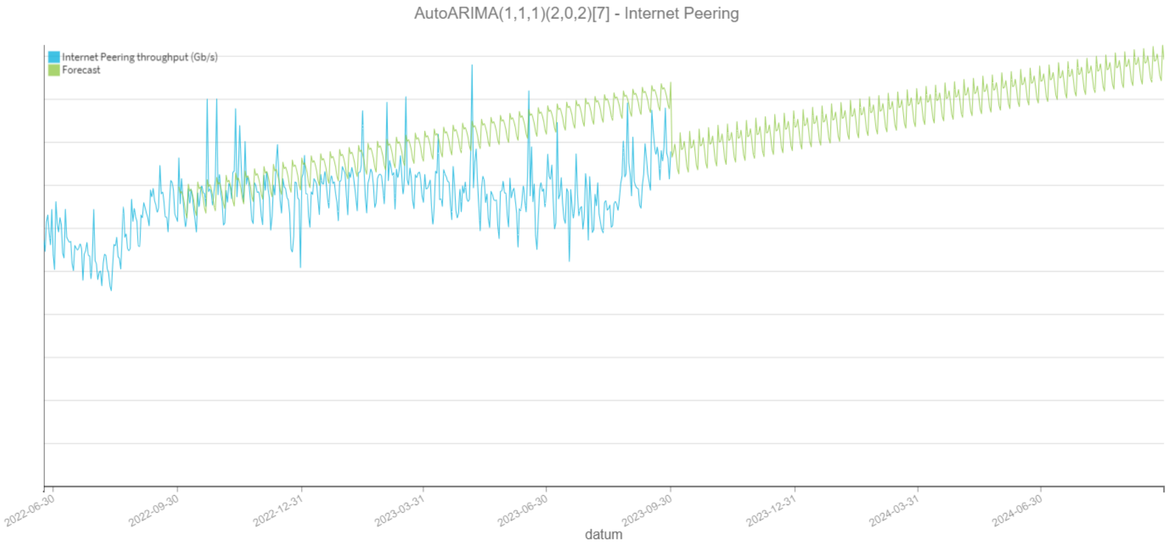


Figure B.1: ARIMA(1,1,1)(2,0,2)[7] model on the daily Internet Peering maximum traffic throughput in DataIku [10].

When additional input related to anticipated events is provided during model training, SARIMA transforms into a SARIMAX model. The following equation allows for the incorporation of exogenous variables, where k is the number of exogenous variables:

$$Y_t = \alpha + \sum_{i=1}^{k} \beta_i X_{i,t} + \dots$$

Here $X_{i,t}$ represents the exogenous variables and β_i the coefficients for the exogenous variables.

Tests with AutoARIMA models have been performed to test whether the MAPE improved when exogenous variables were used as input. Table B.1 shows the results of these tests. Only for the models with a longer forecast horizon than 31 days, the results improved. The improvement in MAPE scores for the AutoARIMA models suggests that these models perform better in capturing underlying trends and patterns when predicting further into the future. The short-term fluctuations tend to smooth out and the exogenous variables have a more significant impact on the forecast when considered over a longer period.

Table B.1: Comparison of AutoARIMA models without and with exogenous variables.

AutoARIMA		**MAPE**							
Service domain	Season length	FH: 7 days	with RD	FH: 31 days	with RD	FH: 183 days	with RD	FH: 365 days	with RD
Data centers	7 days	2,50%	2,83%	6,25%	6,38%			12,90%	8,20%
	14 days	2,80%	3,26%	6,18%	6,53%			8,40%	
Internet Peering	7 days	5,60%	6,53%	6,53%	6,78%			10,10%	8,80%
	14 days	5,90%	6,81%	6,73%	7,03%			10,20%	
Mobile Core	7 days	2,90%	3,90%	6,41%	6,48%	8,50%	8,10%	-	
	14 days	3,10%	3,37%	6,72%	6,38%			-	
Wholesale	7 days	4,43%	4,85%	5,81%	5,89%			29,40%	30,00%
	14 days	4,85%	4,99%	6,08%	6,13%				
Transport Core	7 days	3,97%	4,48%	5,42%	5,70%			13,70%	10,40%
	14 days	4,23%		5,52%					

B.2. DeepAR

DeepAR is based on the architecture of an RNN cell. The equation of a simple RNN using the hidden state $h(t)$ is expressed as follows [88]:

$$h(t) = f(h(t-1), x(t); \theta) \tag{B.1}$$

Equation B.1 shows the previous hidden state $h(t-1)$, the input $x(t)$ at time step t and θ, represents the parameter of the transition function f. The hidden state at the current time step is updated based on the previous hidden state.

Min-max rescaling is a data preprocessing technique that scales input features to a specified range. As the input column is based on 0 and 1, min-max rescaling is used to ensure uniform influence on the model. Moreover, the epoch argument is set. The number of epochs determines the number of iterations the data is passed through the neural network [89]. The most accurate model is with seven epochs. This results in a MAPE of 9.7% and the prediction results are depicted in Figure B.2.

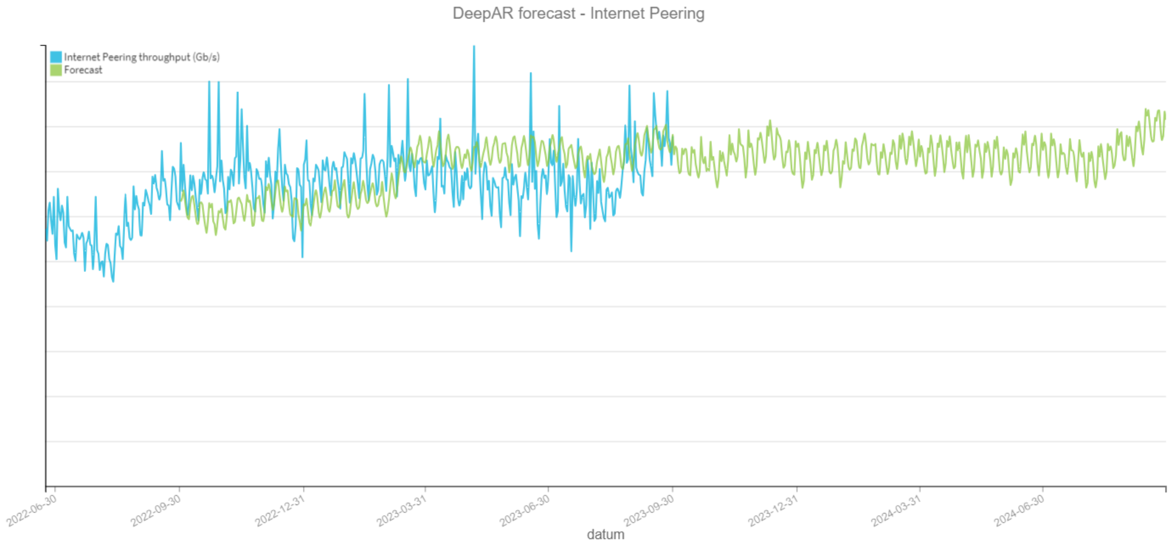


Figure B.2: The results of the DeepAR model with the Internet Peering time series as input [10].

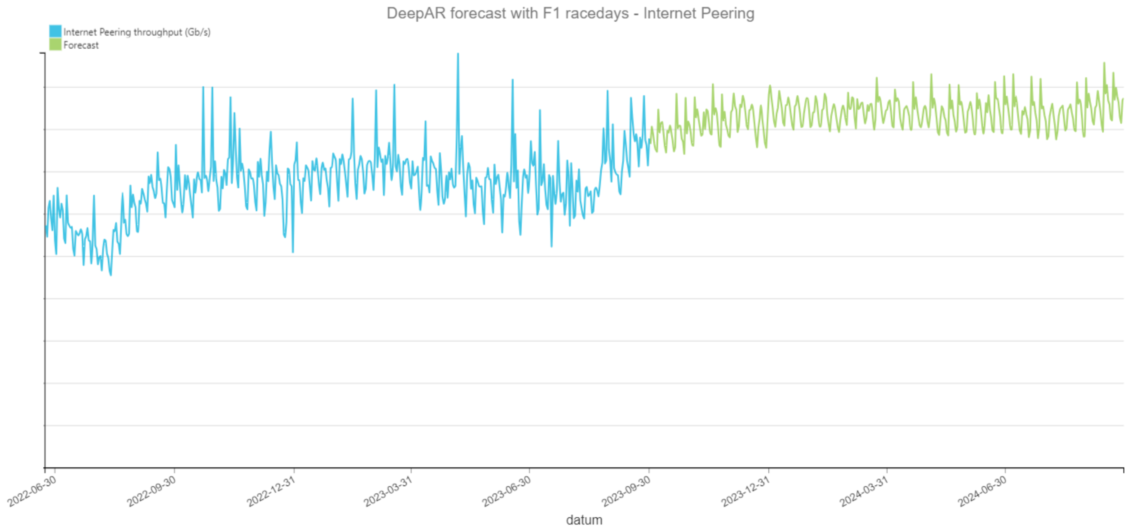


Figure B.3: The results of the DeepAR model with the Internet Peering time series and exogenous variables as input [10].

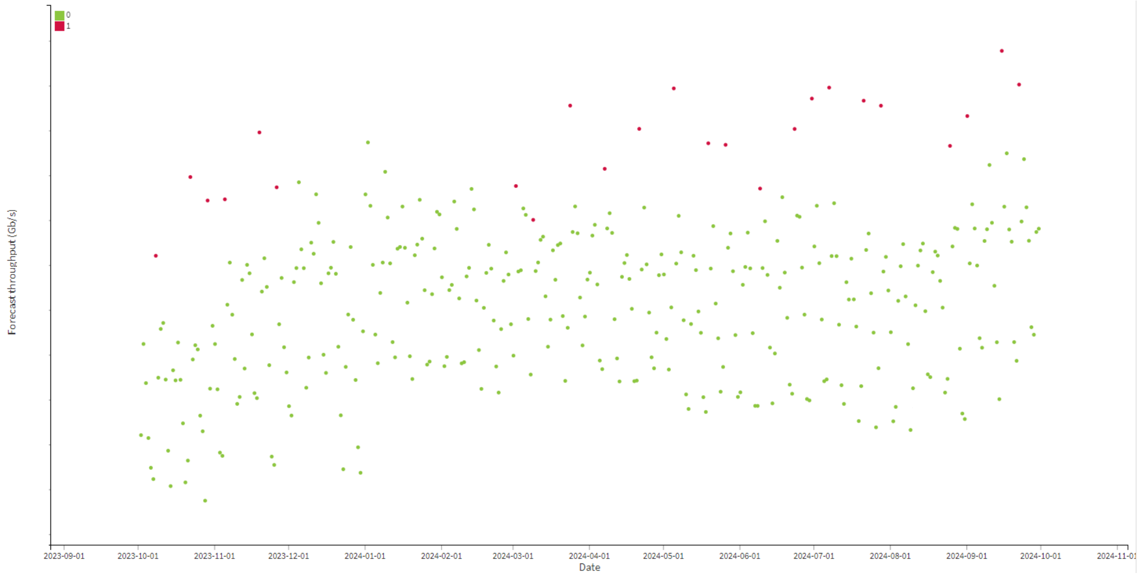


Figure B.4: The forecast values on race days (1) and not on race days (0) [10].

B.3. Decomposed forecasts

B.3.1. Internet Peering

Figure B.5 shows the seasonal naive forecast of the seasonal component of Internet Peering. The MAPE of the seasonal naive forecast is 0.0%.

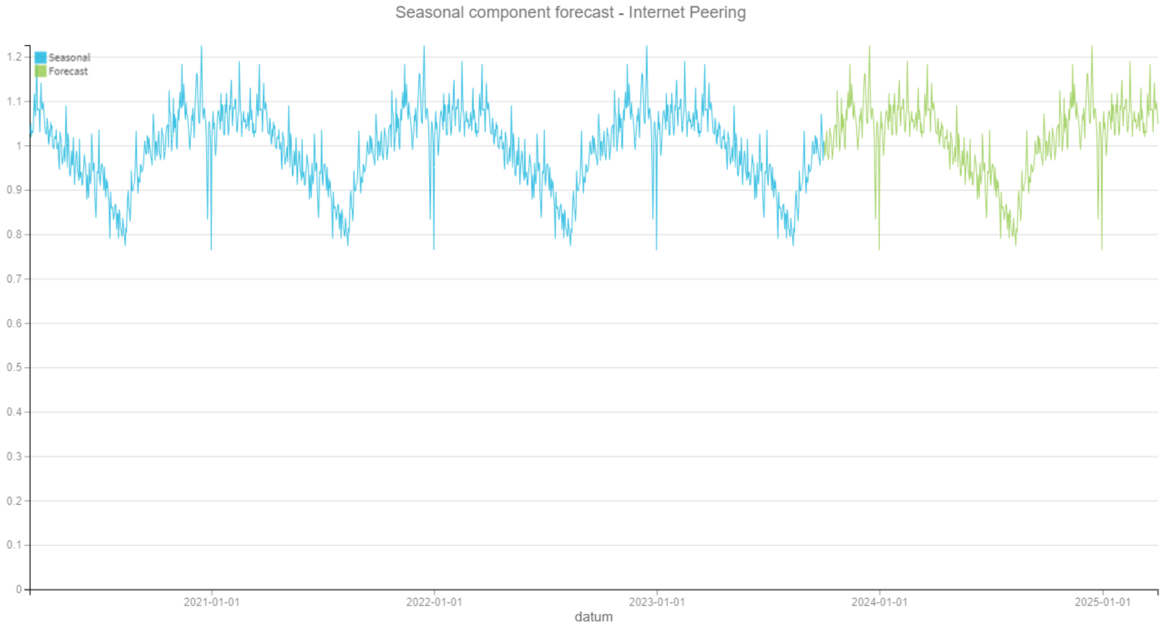


Figure B.5: The seasonal naive time series forecast of Internet Peering [10].

The trend component can be forecasted using a regression technique, like ARIMA. The ARIMA model that resulted from the trend is shown in Figure B.6.

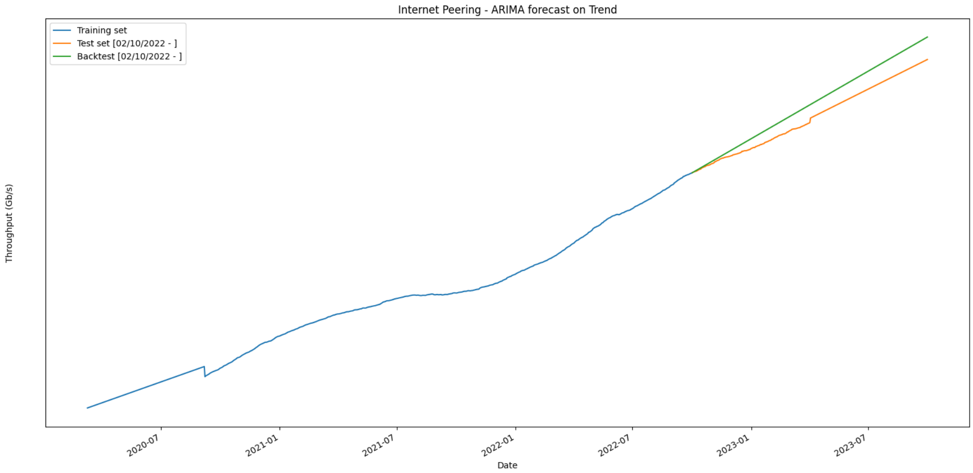


Figure B.6: ARIMA(0,2,1) model fitted on the trend of Internet Peering.

The MAPE of the ARIMA(0,2,1) model is 2.0%. The trend is differenced twice and uses a moving average model in order one to obtain the forecast. This model is then employed to forecast the time series for 1,5 years in advance. The results of this forecast can be seen in Figure B.7.

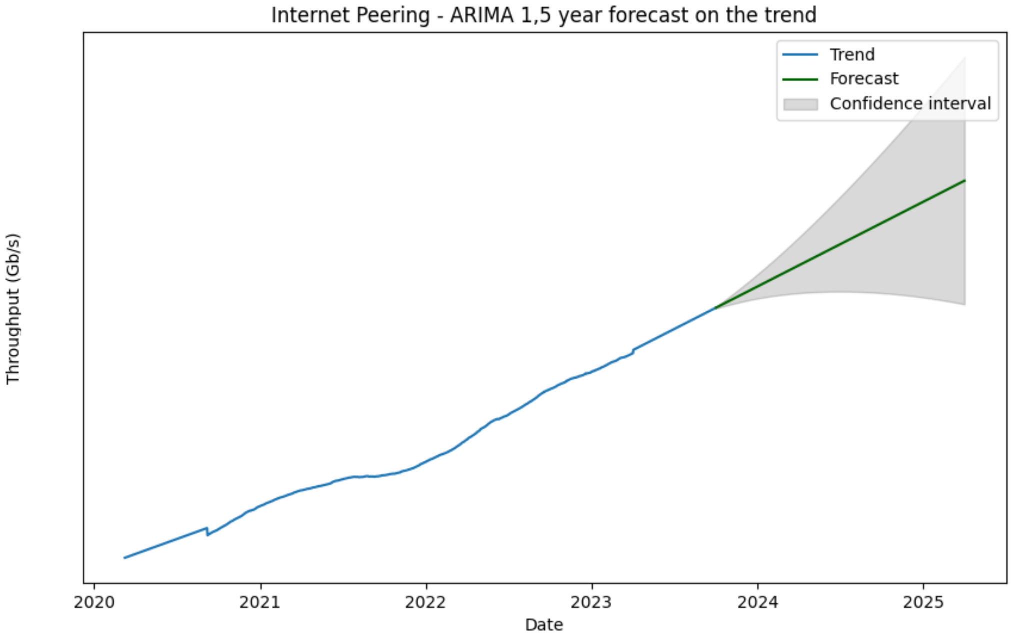


Figure B.7: ARIMA(0,2,1) forecast on the trend of Internet Peering.

These values are used for the forecast of the trend component in the decomposed forecast.

B.3.2. Mobile Core

The seasonal naive forecast of the seasonal component of Mobile Core is illustrated in Figure B.8. The MAPE of the seasonal naive forecast is 0.4%.

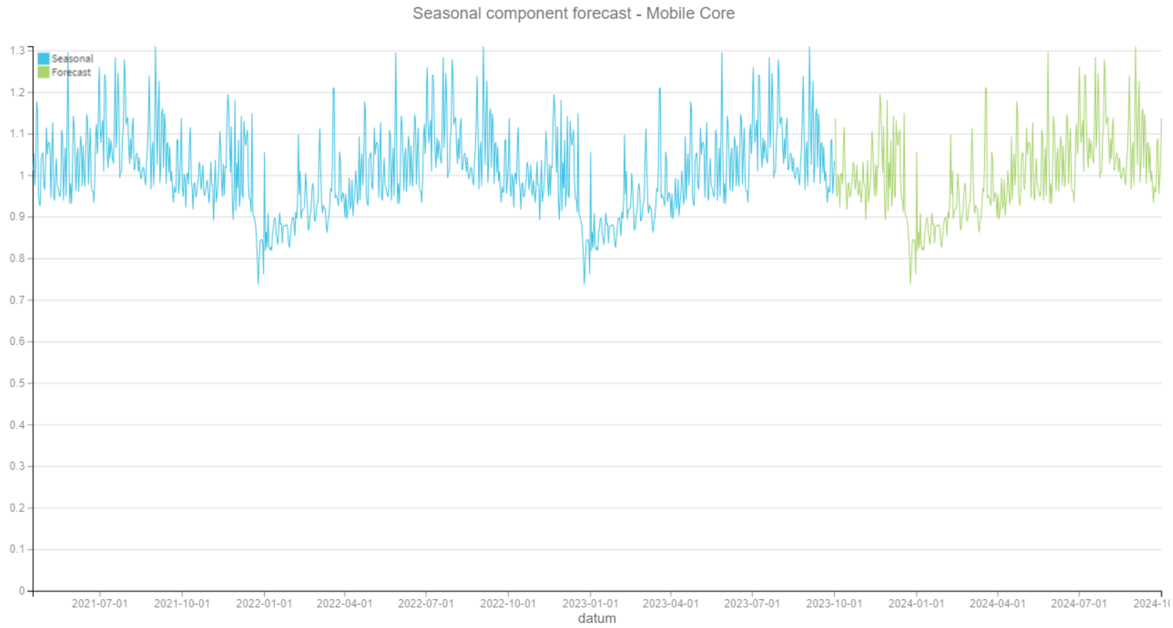


Figure B.8: The seasonal naive time series forecast of Mobile Core [10].

The ARIMA forecast of the decomposed trend resulted in an ARIMA(0,2,1) model, which is depicted in Figure B.9. The MAPE of the ARIMA(0,2,1) model is 0.1%.

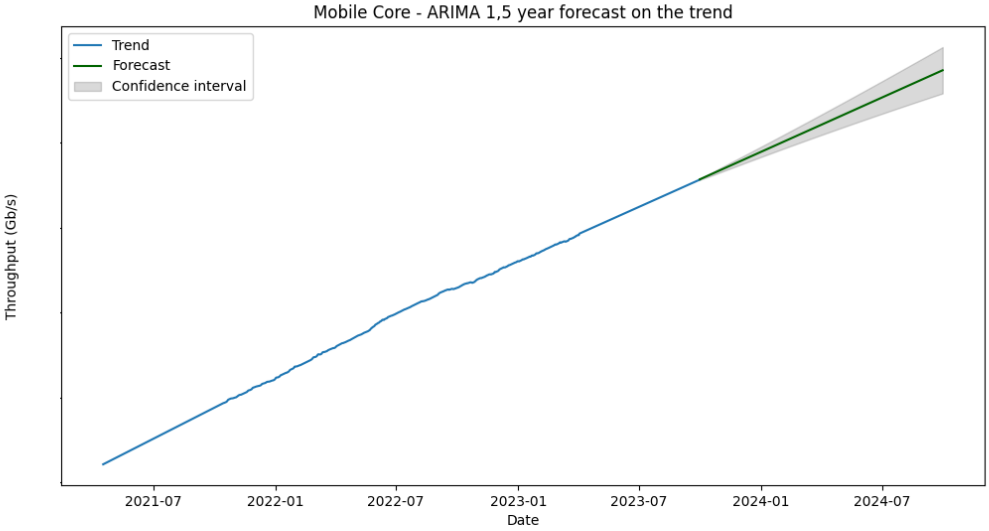


Figure B.9: The decomposed trend forecast with ARIMA(0,2,1).

C
Extreme Value Analysis

Figure C.1 presented below, depicts the Block Maxima method applied to the Internet peering time series. The blocks have been set to a size of 31 days, where the maximum value per 31 days is considered as the extreme for that block. However, some blocks do not contain an outlier compared to the throughput at that moment. This is for instance shown in the block before the end of 2020. Despite this, the Block Maxima approach still considers the maximum value in the block as an extreme value. Additionally, some blocks may have multiple extreme values, but only the maximum value is considered for the EVA model. This may result in a loss of information and an inaccurate representation of the extremes in the dataset. Therefore, the Block Maxima approach is not chosen for the EVA model.

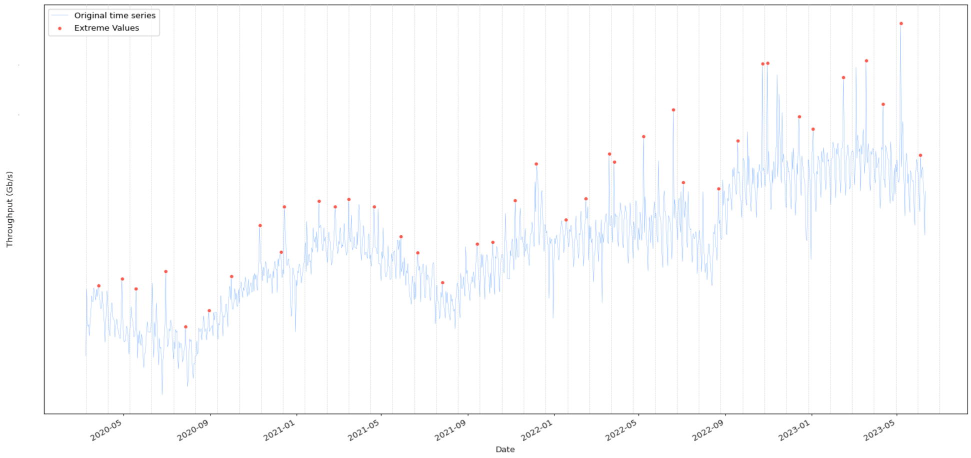


Figure C.1: The Block Maxima threshold approach on the Internet peering time series.

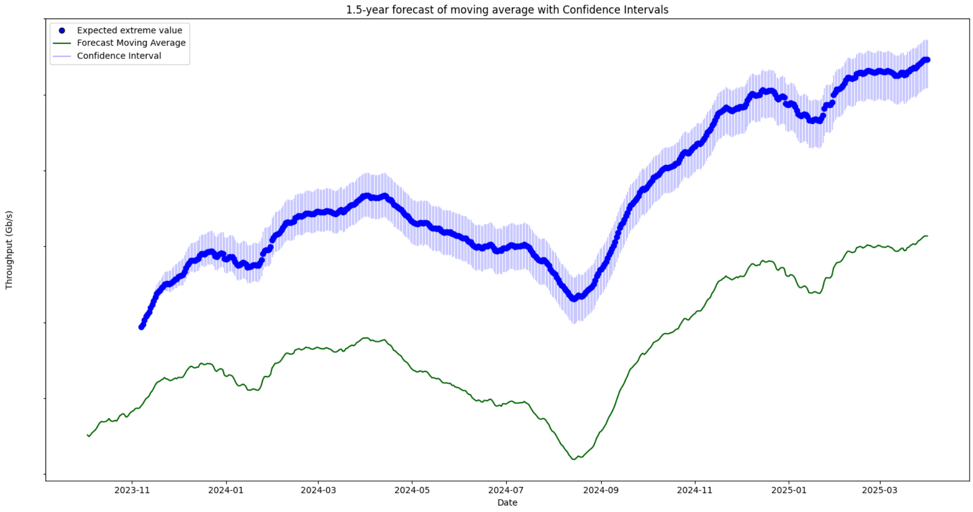


Figure C.2: The results of the EVA model for 6 quarters, with $\alpha = 95\%$.

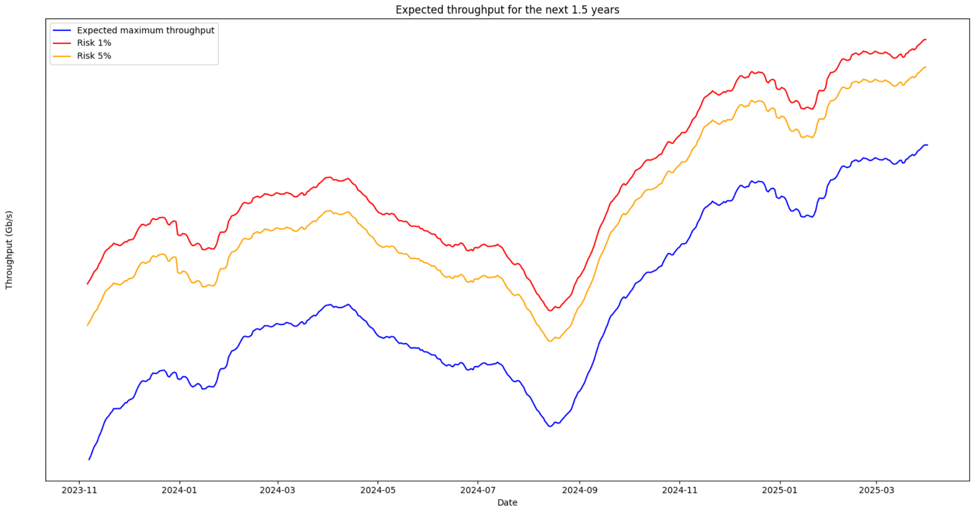


Figure C.3: The predicted results of the prediction framework.